

TEMA 1

- Introducción a los S.I.

1.1 Definición del sistema: Es un conjunto de cosas que ordenadamente relacionadas entre sí contribuyen a un determinado objetivo. Elementos:

- Componentes del sistema
- Relaciones entre los componentes
- Objetivo del sistema
- El entorno del sistema
- Los límites del sistema o frontera entre el sistema y lo que constituye su entorno.

1.1.1. Enfoque sistémico u holístico: Es la manera de estudiar o analizar sistemas partiendo de una visión global y se van descomponiendo de arriba abajo.

Inicialmente el sistema global pueden distinguirse sus entradas y salidas pero no su interior, una vez identificados se describe su interior y se identifican los subsistemas que componen el sistema y las relaciones entre ellos. Esto se utiliza para grandes aplicaciones y lo hacen los analistas.

1.2. Definición de Información: No hay que confundir dato con información, los datos están constituidos por los registros de los hechos,... y la información implica que los datos estén procesados para que sean útiles y significativos para el receptor.

1.3. Sistema de Información: Es una estructura compuesta por un conjunto de personas distribuida en departamentos o funciones con arreglo a ciertos criterios de división de trabajo y coordinación.

1.3.1 Definiciones de S.I. basados en sus objetivos: Es ayudar al desempeño de las actividades en todos los niveles de la organización mediante el suministro de la información adecuada con la calidad suficiente a la persona apropiada en el momento oportuno y con el formato más útil para el preceptor.

1.3.2. Los Elementos de un S.I. : El S.I. debe estar por tanto al servicio de los objetivos generales de la empresa, los individuos de la empresa adoptan las formas de trabajo más útiles y eficaces, las prácticas de trabajo determinan qué información se necesita, cómo se ven implicadas las personas y qué equipo se precisa para un trabajo eficaz.

- Procedimientos y prácticas habituales de trabajo
- La información
- Las personas o usuarios q usan la información para realizar sus actividades.
- El equipo de soporte para la comunicación

1.3.3. La estructura de un S.I:

–Nivel de Operaciones y transacciones: incluye el procesamiento de las actividades diarias. Las transacciones constituyen la mayor parte de las actividades cotidianas. Características:

- Procedimientos de tratamiento se comprenden bien y pueden describirse en detalle.
- Aparecen pocas excepc. a los procedimientos normales.
- Muchas transacciones.

–Nivel Operativo: Es responsable del análisis de los resultados de los recursos consumidos en las transacciones, características:

- Es repetitiva.
- Está centrada en el pasado.
- Los datos tienen un formato bien estructurado (precisos).

–Nivel táctico: La asignación efectiva de recursos a medio plazo, generalmente 1 año para mejorar el rendimiento de la empresa. Se concentra en el análisis de informes de los tipos: – Resúmenes con medidas estadísticas.

– Excepciones para medidas estadísticas q se apartan de la media. – Específicos que no se habían pedido anteriormente.

–Nivel Estratégico: Trabaja a largos plazos (3–5 años), define las líneas maestras que debe seguir la empresa en el futuro.

1.3.4. Flujos de información en la empresa: Hay 3 tipos:

- Flujos verticales ascendentes: Consisten en informes + o – formales sobre resultados sobre las actividades y avisos de incidencias.
- Flujos verticales descendentes: Son ordenes, decisiones y objetivos que hay que cumplir
- Flujos horizontales entre personas del mismo nivel: consiste en informaciones de coordinación para poder manejar responsabilidades compartidas y adaptarse mejor al mercado.

1.4. Otros conceptos relacionados con los S.I.:

–S.I para la gestión: Se dedica a los niveles Operativo, táctico y estratégico de dirección. Es proporcionar a los directivos la información para tomar las decisiones estructuradas.

–Sistema de apoyo a las decisiones: Debe dar soporte a las decisiones poco estructuradas

–Sistemas de procesamiento de transacciones: Dedicada al tratamiento de las operaciones rutinarias, diarias o transacciones.

- Sistemas de Información básicos en las empresas

2.1.Subsistema de Recursos Humanos: Se encarga de la gestión del personal y de la gestión de la nómina.

–Nivel Operativo: –Mantenimiento de los datos de los empleados

–Inventario de puestos de trabajo existentes en la empresa. – Evaluación de los empleados

– Nivel Táctica: –Análisis y diseño del perfil de persona ideal para cada puesto de trabajo. –Generar planes para aplicar beneficios sociales a los empleados.

–Nivel Estratégico: –Planificar la cantidad de personal

–Planes de jubilación

–Reestructuración de la plantilla en general

2.2. Subsistema de Gestión Comercial: Se encarga de las ventas, comercialización y las compras. Actividades:

- Las ventas dentro :-Gestión y tratamiento de los pedidos
- La facturación de la venta o pedido
- El control de los detalles de entrega
- La comercialización o marketing implica el análisis de las ventas de la competencia, de los gustos y demandas de los clientes:

- Estadísticas sobre las ventas
- Investigación del mercado
- Datos sobre la capacidad financiera de la empresa

-Nivel Operativo: Gestión de proveedores, Gestión de pedidos, Gestión de facturas, Ofertas de producto. Apoya a los vendedores

-Nivel Táctico: Recogida de información de ventas, Gestión de publicidad...

-Nivel Estratégico: Seleccionar los segmentos del mercado a los que se quiere acceder; Planificar los productos y servicios q se deben ofrecer; Predecir los productos en venta.

2.3. Subsistema de Gestión Contable y Financiera:

-Nivel Operativo: Gestión Activa, Gestión Pasiva, Control de inventarios, Generar informes, Anotaciones en los libros de contabilidad, ejecución de la nomina.

-Nivel Táctico: Inventario de fin de ejercicio; Auditorias: Balance de situación; Elaboración y control de presupuestos.

-Nivel Estratégico: Obtención de las previsiones financieras a largo plazo

2.4. Subsistema de Almacén o control de las existencias: Se encarga del lugar físico donde se guardan los productos. Niveles:

-Nivel Operativo: Control de Entradas, Control de Salidas, Control de Stocks.

-Nivel Táctico: Realizar inventarios, balance anual de desperfectos.

-Nivel Estratégico: Planes de almacenaje a largo plazo.

2.5. Servicios informáticos en una empresa: Vienen suministrados por e centro de procesos de datos (CPD); es el conjunto de recursos físicos y humanos necesarios para la organización, realización y control de las actividades informáticas de la empresa. Tiene 2 partes:

- Explotación o centro de cálculo; lugar donde se trata la información.
- Datos y desarrollo: donde se elabora el software.

2.5.1. Funciones de un CPD :

-Desarrollo de sistemas Informáticos pasando por todas sus fases: Análisis, diseño y programación del

software, pruebas, implantación y mantenimiento.

–Explotación: Tratamiento de los datos para obtener resultados.

–Soporte técnico: Mantenimiento del sistema.

–Organización general: Supervisión, planificación y control de proyectos; Gestión financiera y de personal...

2.5.2. Localización del CPD: Hay dos tipos de CPD:

–CPD centralizado: Existe uno en la empresa. –Independientes: solo dependen de la dirección.

–Dependientes: dependen alguna función de la empresa.

–CPD descentralizado: Está repartido físicamente en diferentes puntos de la empresa.

2.5.3. Tipos de CPD según su estructura organizativa:

–Organigrama Funcional del CPD: el CPD posee tantos departamentos como fases hay en el S.I

–Organigrama por proyectos del CPD

2.5.4. Personal del CPD: Director del CPD, Jefes de proyecto, Analistas, Programadores, Jefe de Explotación o del centro de cálculo, Operadores, Técnicos de sistemas, Administradores de la base de dato

TEMA 2

• Ciclo de Vida del Software (CVS)

1.1. Concepto de ciclo de Vida: Es el conjunto de fases por las que pasa el software a lo largo del tiempo desde que se empieza a construir hasta que se deja de usar. El ciclo de desarrollo del Software al conjunto de fases por las que pasa el software a lo largo del tiempo desde la fase de análisis hasta la entrega del sistema al usuario.

1.2. Procesos del CVS:

1.2.1. Procesos principales: Son aquellos q resultan útiles a las personas que inician o realizan el desarrollo , el mantenimiento del software durante su CVS.(compradores, suministradores,...)

- *Proceso de Adquisición*: Contiene actividades que el cliente realiza para adquirir el producto Software.
- *Proceso de Suministro*: Contiene las actividades que realiza el suministrador
- *Proceso de desarrollo*: Contiene todas las actividades de desarrollo del software. Actividades: Análisis de los requisitos del sistema, Diseño de la arquitectura de alto nivel del sistema, Análisis de los requisitos del Software, Diseño de la arquitectura del Software, Diseño detallado del software, Codificación y prueba del Software, Integración del sistema, Prueba del sistema , Instalación del software , Aceptación del Software por parte del comprador.
- *Proceso de Explotación u operación*
- *Proceso de Mantenimiento*; Cuando El software necesita modificaciones debido a un error.

1.2.2. Procesos de Soporte:

- *Proceso de Documentación*

- *Proceso de Gestión de Configuración*: Aplica procedimientos administrativos y técnicos durante todo el CVS
- *Proceso de Aseguramiento de Calidad*: El Software cumple con los requisitos especificados
- *Proceso de Verificación*: Verifica si los requisitos del sistema están completos y son correctos
- *Proceso de Validación*: Si se usa el Software para lo que fue construido
- *Proceso de Revisión Conjunta*: Evalúa el funcionamiento del sistema
- *Proceso de Auditoria*: Si se han cumplido los requisitos, los planes y el contrato.
- *Proceso de Resolución de Problemas*: Analiza y elimina los problemas descubiertos en el software.

1.2.3. Procesos Generales:

- *Proceso de Gestión*: Actividades de planificación, revisión, evaluación
- *Proceso de Infraestructura*: Hardware, Software, herramientas, normas, técnicas e instalaciones para el desarrollo
- *Proceso de Mejora*: Controla y mejora los procesos del CVS.
- *Proceso de Formación*: Proporciona formación y mantiene al personal formado.
- Procesos de Adaptación.
- Modelo del CVS:

1.3.1. Modelo en Cascada o Waterfall: Modelo lineal secuencial, es el modelo más antiguo pero más utilizado. Características:

VENTAJAS:

- Cada fase empieza cuando ha terminado la fase anterior
- Previene que se sobrepasen las fechas de entrega y los costes esperados.
- Pasar de una fase a otra es preciso conseguir todos los objetivos de la etapa previa
- Es el preferido de los jefes de proyecto

INCONVENIENTES:

- ♦ Es lineal y el desarrollo del Software no lo es
- ♦ Los proyectos no siguen el flujo secuencial que propone el modelo
- ♦ A menudo es difícil que el cliente exponga todos los requisitos al principio
- ♦ Tarda mucho tiempo en recorrer todo el ciclo
- ♦ Un error grave puede ser desastroso si no se detecta hasta que se revisa el programa

1.3.2. Modelo de construcción de prototipos: Este modelo comienza con la definición por parte del cliente y del desarrollador de los objetivos globales y la identificación de los requisitos conocidos, entonces el desarrollador realiza un diseño que será visible para el usuario. El cliente evalúa el prototipo y lo utilizan para refinar los requisitos del Software a desarrollar.

Ventajas: Mejora la comunicación desarrollador–cliente, satisface la curiosidad del cliente porque enseguida puede ver cosas.

Desventajas: Identificación del prototipo con el producto final, el prototipo no es reprovechable.

1.3.3. Modelo de Procesos Evolutivos del Software: Define los detalles de extensiones del mismo. Hay dos modelos:

- ♦ Modelo Incremental: Combina elementos en cascada con la interactiva de construcción de prototipos. El sistema Software se va creando añadiendo componentes funcionales al sistema llamados *incrementos*. Este modelo es interactivo, se centra en la entrega de un producto operacional con cada incremento. Los primeros incrementos son versiones del producto final, pero proporcionan la capacidad que sirve al usuario así como una plataforma para la evaluación por parte del usuario.

–Ventajas: Satisface la curiosidad del cliente, posibilidad de detener el proyecto sin perder todo lo realizado

–Desventajas: El diseño y desarrollo debe ser coherente entre si, control del proyecto sin poderlo ver todo.

- ♦ Modelo en Espiral: El software se desarrolla en una serie de versiones incrementadas durante las primeras iteraciones. Este modelo se divide en regiones de tareas:

–Comunicación con el cliente: Contiene las tareas requeridas para establecer comunicación entre el cliente y desarrollador.

–La Planificación: Contiene las tareas requeridas para definir los recursos, el tiempo y otras informaciones relacionadas con el proyecto.

–Análisis de riesgo: Contiene las tareas requeridas, para evaluar los riesgos técnicos y de gestión

–Ingeniería: Contiene las tareas requeridas para construir una o más representaciones de la aplicación (prototipos).

–Construcción y adaptación: Contiene las tareas requeridas para construir, probar, instalar y proporcionar soporte al usuario.

–Evaluación del cliente: Contiene las tareas requeridas para obtener la reacción del cliente según la evaluación de las representaciones del Software creadas durante la etapa de ingeniería e implementada durante la etapa de instalación.

En este modelo cada ciclo se completa con una revisión en la que participan las principales personas que tienen relación con el producto.

El primer circuito de la espiral produce el desarrollo de una especificación de productos, los pasos siguientes en la espiral se podrían utilizar para desarrollar un prototipo y progresivamente versiones más sofisticadas del software.

Utiliza la construcción de prototipos como mecanismo de reducción de riesgos.

Ventajas: Proceso evolutivo de desarrollo de SW, Uso de prototipos o simulaciones, El cliente ve evolucionar el proyecto.

Desventajas: Es un método muy nuevo, necesita gran habilidad en la elaboración de riesgos y es difícil convencer al cliente de su utilidad.

2. Metodologías de desarrollo del Software

2.1. Introducción: Las metodologías son un conjunto de pasos y procedimientos a seguir para el desarrollo del Software. En general las metodologías pretenden: Obtener aplicaciones mejores, mejorar el proceso de desarrollo, tener un proceso estándar en la organización.

La diferencia entre metodología y ciclo de vida es: El ciclo de vida indica qué es lo que hay que obtener a lo largo del desarrollo del proyecto, pero no cómo, esto si lo debe indicar la metodología. La metodología puede seguir uno o varios modelos de Ciclo de vida.

Visión histórica: –Desarrollo convencional , –Desarrollo Estructurado , –La programación estructurada determina como había que ver un programa de forma que fuera lo más comprensible posible, – El diseño estructurado, – El Análisis Estructurado, –Modelos físicos actuales que representan los diferentes procesos que realizan el sistema actual y que indican detalles de cómo interactúan los usuarios sobre los mismos y dónde se localizan dentro de la organización, – Modelos lógicos actuales representan los procesos funcionales a un nivel abstracto, – Desarrollo orientado al objeto trata procesos y datos de forma conjunta.

2.2 Clasificación

2.2.1. Metodologías Estructuradas: Proponen la creación de modelos del sistema que representan los procesos, los flujos y la estructura de los datos de una manera descendente. Hay dos tipos:

- ◆ Orientada a Procesos: Se basan en la utilización de un método descendente de descomposición funcional para definir los requisitos del sistema, dan lugar a un nuevo concepto que es la especificación estructurada a es un modelo gráfico particionado, descendente y jerárquico de los procesos del sistema. *Se compone de:* –*DFD*: Son diagramas que representan los procesos de datos que deben llevar a cabo un sistema a distintos niveles de abstracción y los datos que hay entre las funciones. –*DD*: Es el conjunto de las definiciones de todos los datos que aparecen en el DFD. –*Especificaciones de procesos*: como se obtienen las salidas del proceso a partir de sus entradas.
- ◆ Metodología de Demarco: Estudio del sistema físico actual, Derivación del modelo lógico actual, derivación al nuevo modelo lógico, creación de modelos físicos alternativos...
- ◆ Metodología de Gane y Sarson: Es parecida a la de Demarco, la diferencia es que elimina el primer paso y crea uno nuevo que es cuando construye el modelo lógico del sistema no solo elabora una especific. Estructur. Sino que también construye un modelo lógico de datos.
- ◆ Metodología de Yourdon: Realiza los DFDs del sistema, A partir de los DFD realiza el diagrama de estructuras , Evaluación del diseño, Preparación del diseño
- ◆ Orientadas a los datos: Se basan o se fijan más en las entradas y las salidas.

2.2.2. Metodologías Orientadas a Objetos: Examinan el sistema como un conjunto de objetos que interactúan entre sí. Hay 2 enfoques:

- ◆ Revolucionarios o puros: Entienden la orientación al objeto como un cambio profundo en la forma de analizar y diseñar
- ◆ Evolutivos: Piensan que el análisis y el diseño estructurado constituyen la base para el desarrollo ambientado al objeto.

2.2.3. Sistemas de Tiempo Real: Son sistemas muy dependientes del tiempo que procesan información más orientada al control que a los datos se controlan por medio de eventos externos. Características: Se lleva a cabo el proceso de muchas actividades de forma simultánea, se asigna prioridades a determinados procesos, existe comunicación entre tareas, existe accesos simultáneos a datos comunes.

2.3. Principales Metodologías

2.3.1. MERISSE: –Ciclo de vida más largo que los existentes

–Introducción de 2 ciclos complementarios:

–C. De decisión

–C. De abstracción: Se basa en la percepción de 3 niveles de abstracción en el q se definen dos modelos: Modelo de datos y Modelo de tratamientos:

–*Nivel Conceptual*: Se ocupa de definir el qué a través de un conjunto de reglas de gestión q pesan sobre la empresa.

–*Nivel Organizativo*: Corresponde a la organización adecuada q hay que implementar en la empresa para alcanzar los objetivos asignados al siste.

–*Nivel Físico u Operativo*: Corresponde a la integración de los medios técnicos para el proyecto.

2.3.2. SSADM: Introduce una técnica llamada técnica del diseño del diálogo que se utiliza para diseñar la interfaz del usuario, los aspectos fundamentales son: –Enfatiza mucho el nivel de usuario, – Analiza y diseña centrándose en 3 aspectos: datos, procesos y eventos, – Aporta máxima flexibilidad.

2.3.3. MÉTRICA: Crea un entorno que permita al equipo de trabajo construir sistemas que: – Den soluciones a los objetivos considerados prioritarios

- ◆ Se desarrollen cuando el usuario los necesite y de acuerdo a los presupuestos y duración estimados
- ◆ Se mantengan fácilmente para soportar los cambios futuros de la organización.

TEMA 3

- Cómo comienza un proyecto: Tiene varias fases:
 - ◆ La empresa decide emprender el proyecto: Define y analiza los requisitos del sistema que es fundamental obtener de los usuarios la información pertinente sobre las necesidades que desean que el software satisfaga. Al comienzo del proyecto los principales proveedores de información son los directivos. Los resultados de estos estudios previos se recogen en un documento llamado informe de necesidades. En este momento comienza el estudio de la viabilidad del proyecto para ello hay que analizar:–Las diferentes alternativas q se pueden concebir, –Evaluar cada una de las alternativas, – Especificación detallada de la alternativa seleccionada, – Establecimiento de fechas y compromisos de trabajo.
 - ◆ Selección del jefe de proyecto: El jefe es el elemento más crítico para su éxito. Las características son: – Liderazgo, Comprensión técnica, Competencia en la gestión, Presteza y decisión, Versatilidad y flexibilidad, Integridad, Ser Previsor.
 - ◆ Comienzo del proyecto: El jefe de proyecto establece el entorno inicial de trabajo, este entorno incluye la identificación de las áreas de gran riesgo del proyecto, etc. También debe definir el soporte necesario para el equipo del proyecto, las técnicas de comunicación entre sus miembros y en su caso los requisitos que deberán cumplir los posibles subcontratistas. Un plan de proyecto debe definir que hay que hacer, como se hará, quien lo hará,...
- Estudios de viabilidad: Hay que tener en cuenta 4 aspectos:
 - ◆ *Económico*: Determinar si el beneficio obtenido compensa los costes.
 - ◆ *Técnico*: Estudiar si la funcionalidad y el rendimiento son realizables
 - ◆ *Legal*: Ver si los requisitos atentan con alguna ley o reglamento
 - ◆ *Operativo*: Ver si se pueden implantar de manera decisiva en la empresa.

Análisis de costes– beneficios: Determina los costes para el desarrollo del proyecto y los compara con los beneficios tanto tangibles como intangibles. Nos permite seleccionar la alternativa más beneficiosa y prever las necesidades financieras. Se representa en forma de tabla, en las columnas aparece los años de vida del proyecto y en las filas los conceptos de gasto y de beneficio.

$$\text{Valor actual} = \text{Beneficio Neto} / (1+t/100)$$

Siendo t la tasa de inflación y n el año de aplicación

- **Técnicas de recogida de Información:** Es un proceso mediante el que se consigue que los usuarios descubran los requisitos que desean en la aplicación. Son un medio para mejorar la comunicación entre usuarios y desarrolladores del Software. El proceso de análisis debe seguir 5 pasos:
 - ◆ Identificar las fuentes de información relevantes para el proyecto
 - ◆ Realizar las preguntas apropiadas
 - ◆ Analizar la información recogida
 - ◆ Confirmar con los usuarios lo que se ha comprendido de los requisitos.

Las técnicas principales utilizadas son:

- ◆ Entrevistas
- ◆ Desarrollo conjunto de aplicaciones
- ◆ Prototipado
- ◆ Observación
- ◆ Estudio de la documentación existente en la empresa
- ◆ Tormenta de ideas o Brainstorming: Reuniones de usuarios en las que en una primera fase se sugieren toda clase de ideas por muy disparatadas que parezcan,
- ◆ Ethics: Se busca la satisfacción de los empleados en el trabajo a través de estudios integrales.

3.1. Las entrevistas: Es un intento sistemático de recoger información de otra persona a través de una comunicación interpersonal que se lleva a cabo por medio de una conversación estructurada. Fases a distinguir:

- ◆ **Preparación:** El entrevistador debe documentarse e investigar la situación de la organización, analizando los documentos de la empresa disponible. Hay que intentar minimizar el nº de entrevistados, hay que considerar las entrevistas de cortesía, analizar el perfil de los entrevistados, Definir el objetivo y el contenido de la entrevista, Planificar el lugar y la hora en la que se va a desarrollar la entrevista es conveniente de realizarla en un lugar confortable. Algunos proponen enviar previamente el entrevistado un cuestionario y un pequeño documento de introducción al proyecto de desarrollo.
- ◆ **Realización:** Hay tres fases: **Apertura:** Presentarse e informar al entrevistado sobre la razón de la entrevista.

Desarrollo: Cumplir las reglas del protocolo, hay que llegar a un acuerdo sobre como se va a registrar la información obtenida. **Terminación:** Se termina recapitulando la entrevista agradeciendo el esfuerzo y dejando abierta la posibilidad de volver a contactar para aclarar conceptos o bien citándole para otra entrevista.

- ◆ **Análisis:** Consiste en leer las notas, pasarlas a limpio, reorganizar la información, contrastarlas con otras entrevistas o fuentes de información, evaluar como ha ido la entrevista,...

3.2. Desarrollo conjunto de aplicaciones (JAD): Promueve la cooperación y el trabajo en equipo entre usuarios y analistas mediante un conjunto de reuniones de varios días de duración, las razones que sirven de base a esta técnica son:

- ◆ Se requiere mucho tiempo para las entrevistas, prepararlas, hacerlas, redactar un conjunto coherente de requisitos
- ◆ Es más difícil cometer errores en la especificación de requisitos porque lo revisa todo el equipo
- ◆ Tiene una mayor probabilidad de éxito porque el usuario se siente involucrado, consta de unas fases: Adaptación o preparación, Sesión JAD: Partiendo de un documento de trabajo se

analiza para completar el conjunto de requisitos del sistema, documentación: Redactar y documentar los detalles, pasarlos a limpio, dar un formato adecuado al texto...

3.3. El prototipado: Elaboración de un modelo o maqueta del sistema que se construye para evaluar mejor los requisitos que desean que cumpla. La cualidad esencial de un prototipo debe ser la posibilidad de ser construido más rápidamente que la aplicación correspondiente. Las razones para emplear el prototipado son 3:

- ◆ Prototipado de la interfaz de usuario: Para asegurar que está bien diseñada, no cuesta mucho.
- ◆ Modelos de Rendimiento: Para evaluar el posible rendimiento de un diseño técnico.
- ◆ Prototipado funcional relacionado con un ciclo de vida iterativo en este caso el prototipo supone una primera versión del sistema con funcionalidad limitada , a medida que se comprueba si las funciones implementadas son las apropiadas se corrigen, refinan o añaden otras nuevas hasta llegar al sistema final.
- ◆ Las herramientas del prototipado pueden ser:
 - Programas de dibujo, de presentación, hojas de cálculo, etc
 - Gestores de bases de datos y sistemas de 4ª generación
 - Herramientas CASE o generadores de aplicaciones: Estas herramientas permiten reutilizar las pantallas, etc para continuar el desarrollo con la misma herramienta.

TEMA 4

1. Introducción: El análisis de requisitos es un proceso del estudio de las necesidades de los usuarios para llegar a una definición de los requisitos del sistema de hardware o software. Fases del análisis:

- ◆ Extracción o determinación de requisitos
- ◆ Análisis de los requisitos obtenidos anteriormente para detectar y resolver posibles inconsistencias o conflictos
- ◆ Construcción de la especificación de requisitos (es un documento que define de forma completa los requisitos)
- ◆ Validación de los requisitos por parte del usuario

2. Especificación de Requisitos del software (ERS): Es la documentación de los requisitos esenciales del software y de sus interfaces externas. Sus características son:

- ◆ Debe incluir información veraz (necesidades reales del usuario)
- ◆ Comunicar la información de forma eficaz
- ◆ Describir qué hay que desarrollar, no debe de escribir ningún detalle del diseño del software excepto las restricciones impuestas en él.

Las características deseables son:

- ◆ No ambigüedad
- ◆ Completa: – Incluya todos sus requisitos significativos del SF
- Defina la respuesta del SF
- ◆ Fácil de verificar: se comprueben que el SF satisface los requisitos
- ◆ Consistente: No contiene requisitos contradictorios o en conflicto
- ◆ Fácil de modificar: Debe tener una organización coherente y manejable con una tabla de contenidos, índices y referencias cruzadas.
- ◆ Facilidad para identificar el origen y las consecuencias de cada requisito
- ◆ Facilidad de utilización durante la fase de mantenimiento y de explotación.

Hay que tener dos consideraciones en cuenta en este proceso:

- ◆ El requisito debe ser especificado de la forma más completa posible
- ◆ Debe iniciarse un proceso formal de cambios para identificar y controlar los cambios proyectados.

3. Clasificación de las técnicas de Especificación: Según la forma de representación:

- ◆ Gráficas: Resaltan la conexión entre los distintos componentes del modelo
- ◆ Textuales: Especifican con más detalle los componentes anteriores
- ◆ Marcos o plantillas: Especifican la información relativa a un componente de un modelo que ha sido declarado en un diagrama. Se representa mediante un formulario.
- ◆ Matriciales: Técnicas de comprobación entre modelos que permiten estudiar las referencias cruzadas entre sus componentes

Según el enfoque de modelización bajo el que se crean los modelos:

- ◆ Función: Estudian el sistema observando principalmente su funcionamiento
- ◆ Información: Estudian el sistema observando principalmente qué datos maneja
- ◆ Tiempo: Estudian la respuesta del sistema ante la aparición de eventos temporales.
- Modelización de funciones o procesos
- Diagrama de flujo de Datos (DFD): Es la técnica más importante del análisis estructurado, visualiza el sistema como una red de procesos conectados entre sí. Un DFD es un diagrama en forma de red que representa el flujo de datos y las transformaciones que se aplican sobre ellos al moverse desde la entrada hasta la salida del sistema. Se apoya en otras 2 técnicas: – Diccionario de Datos , – Especificaciones de procesos.

4.1.1. Componentes de un DFD:

- ◆ **Procesos o funciones:** Son componentes funcionales del sistema, representan una función que transforma flujos de datos de entrada en flujos de datos de salida. Cada función tendrá un nº y un nombre. El nombre debe ser representativo y breve. Se representa con un círculo en cuyo interior está el nombre y el nº.
- ◆ **Almacenes de datos:** Representan información del sistema almacenada de forma temporal por tanto representan datos en reposo; deben tener un nombre representativo, su representación gráfica son dos líneas paralelas con el nombre en medio.
- ◆ **Entidades Externas:** Representa el origen o el destino de la información del sistema. Los flujos que parten o llegan a ellas definen el interfaz entre el sistema y el mundo exterior. Su representación gráfica es un rectángulo o cuadrado con el nombre.
- ◆ **Flujos de Datos:** Representan datos en movimiento. Los flujos de datos que se mueven hacia y desde almacenes simples no necesitan nombre si transportan toda la información del registro. Cuando se lee o escribe una porción de los elementos se debe especificar el nombre en el flujo. Tipos: F. De entrada, F. de salida, F. De dialogo

La conexión Entidad Externa– Almacén y viceversa solo es posible con almacenes externos q sirven de interfaz entre el sistema y una entidad externa.

La conexión entre procesos mediante un flujo de datos es posible siempre y cuando el proceso destino comience cuando el proceso origen finaliza. Se pueden dividir en: – Flujo síncrono

–Flujo asíncrono

Los procesos pueden introducir o recuperar datos en los almacenes:

- ◆ **Flujo de consulta:** Indica la utilización de la información del almacén con el proceso.
- ◆ **Flujo de la actualización:** Indica que el proceso va a alterar la información del almacén.
- ◆ **Flujo de diálogo:** Representa como mínimo un flujo de consulta y uno de actualización que

no tienen relación directa.

4.1.2. Características de un buen DFD:

1º Procesos: – Cada proceso debe tener E/S

–El proceso debe ser capaz de generar los flujos de salida a partir de los de entrada

–El proceso no crea datos nuevos solo los transforma

–El proceso no debe perder información

2º Almacenes: – Cada dato que sale primero debe entrar

–No crea datos nuevos

4.1.3. Explosión de un DFD en niveles: Un DFD no debe tener demasiados procesos, para tratar sistemas complejos se utiliza la descomposición que da como resultado la aparición de DFD a distintos niveles. Partiendo de una función de nivel superior que representa el sistema completo se va descendiendo a niveles más detallados mediante un razonamiento Top–Down hasta llegar a niveles que ya no vamos a subdividir.

4.1.3.1. Diagrama de Contexto: Se le conoce como diagrama de nivel cero, representa el sistema de forma global, solo pueden aparecer entidades externas, flujos de datos y un único proceso que representa el sistema en su conjunto. Pueden aparecer almacenes de datos cuando son compartidos entre nuestro sistema y el exterior.

4.1.3.2. Diagrama del sistema: Se le conoce como diagrama de nivel 1, representa las funciones principales a realizar así como la relación entre ellas. Las funciones de este diagrama sean independientes entre sí porque esto facilita la descomposición de cada una por analistas diferentes.

4.1.3.3. Procesos o funciones Primitivas: Son procesos que no se van a descomponer en más diagramas de nivel inferior, bien porque no se puede o bien porque no interesa.

4.1.3.4. Consistencia entre niveles, Reglas de Balanceo: Balancear un modelo de procesos es comprobar que la información que entra y sale de un proceso de nivel n es consistente con la información que entra y sale del DFD en el que se descompone.

4.1.3.5. Otras consideraciones:

- ◆ Descomposición paralela de flujos de datos
- ◆ Convenciones a la numeración
- ◆ Balanceo de los almacenes de datos

4.2. Diccionario de Datos: Es una lista organizada de todos los datos pertinentes al sistema con definiciones precisas y rigurosas para que tanto el usuario como el analista tengan un entendimiento común de todas las E/S, componentes de almacenes y cálculos intermedios.

El DD describe por tanto el significado de los flujos de datos y de los almacenes de datos, presentes en todo el conjunto de DFD. Puede crearse con una herramienta CASE, aunque la mayoría contienen:

- ◆ Nombre del elemento que se describe
- ◆ Alias
- ◆ Donde se usa, como se usa

- ◆ Descripción del contenido

- ◆ Información adicional

4.2.1. Definición de flujos de datos: Siguen una aproximación top-down. Los *almacenes* se definen como entidades repetitivas de datos o grupos de datos. Un *alias* es un sinónimo de una entrada del DD, no es conveniente la utilización de alias porque se crean redundancias en la especificación estructurada.

4.3. Especificación de procesos o mini especificación: Define el procedimiento que realiza un proceso primitivo. Se puede usar cualquier método que satisfaga estas dos condiciones:

- ◆ La EP debe expresarse de manera que pueda ser verificada tanto por el usuario como por el analista.

- ◆ La EP debe especificarse de forma que sea entendible por cualquier involucrado en el sistema.

Las herramientas utilizadas para realizar una EP: Lenguaje estructurado, árboles de decisión, tablas de decisión, diagramas de acción, Pre/Post condiciones.

4.3.1. Lenguaje Estructurado: Es un lenguaje formado por un subconjunto de palabras del idioma de las que unas se utilizan para formar las construcciones propias de la programación estructurada y otras que incluyen un conjunto de verbos que reflejan acciones simples. Es el pseudo código.

4.3.2. Diagramas de Acción: Utiliza niveles de corchetes anidados que representan la estructura lógica utilizando para transformar los datos de entrada en datos de salida. Se dividen en secuencial, alternativa, repetitiva

4.3.3. Árboles de decisión: Es una representación en forma de árbol de los valores de las variables y las acciones tomadas para cada valor así como el orden en que se realiza la decisión se suelen usar cuando el nº de condiciones no es muy grande.

4.3.4. Tablas de decisión: Cuando el proceso debe producir alguna salida o tomar alguna acción basada en decisiones complejas, basadas en distintas variables que pueden tomar diversos valores puede ser aconsejable usar una tabla de decisión. Muestran la función de forma tabular, se crea listando en las filas todas las variables relevantes y todas las acciones relevantes, en las columnas se listan cada una de las posibles combinaciones de las variables. Las columnas reciben el nombre de reglas

4.3.5. Pre/Post condiciones: Describen la función que debe realizar el proceso sin decir mucho acerca del algoritmo que se utilizará.

Las pre-condiciones describen todas las cosas si las hay que deben darse antes de que el proceso pueda comenzar a ejecutarse. Describirán:

- ◆ Q entradas son necesarias

- ◆ Q relación debe existir entre las entradas

- ◆ Q relaciones deben existir entre entradas y almacenes de datos

- ◆ Q relaciones deben existir entre diferentes almacenes o dentro de un almacén dado.

Las post- condiciones describen lo que debe darse cuando el proceso ha concluido. Describirán:

- ◆ Las salidas que generará el proceso

- ◆ Las relaciones que existirán entre los valores de salida y los de entrada

- ◆ Las relaciones que existirán entre los valores de salida y los valores de uno o varios de los almacenes

- ◆ Los cambios que se hayan dado en los almacenes

4.4. Diagramas de descomposición funcional (DDF): Es la descomposición de funciones de un sistema, se usa para representar funciones también sirven para representar otros tipos de información. Hay 3 tipos:

- ◆ Representan las funciones del sistema a diferentes niveles pero no los flujos de datos entre ellas
- ◆ Representa las funciones y sus estructuras de datos de E/S
- ◆ Representan las funciones y los datos de E/S mediante axiomas matemáticos.

4.5. Comprobaciones a realizar sobre la especificación estructurada: Una vez realizada la especificación estructurada hay que revisarla, hay que comprobar:

- ◆ Si los modelos contenidos son completos
- ◆ Si no existen contraindicaciones ni inconsistencias entre los distintos modelos
- ◆ Si los modelos cumplen los requisitos del usuario
- ◆ Calidad: El estilo, la legibilidad, la facilidad de mantenimiento de los modelos producidos.

TEMA 5

• Modelo de Datos : Introducción

El Modelo Entidad /Relación fue desarrollado por Chen en 1976. Es un modelo muy utilizado en el campo de diseños de bases de datos. Su principal ventaja es que es traducible casi automáticamente a un esquema de bases de datos bajo modelo relacional.

• Elementos del Modelo Entidad/ Relación

El MER tiene sus propias estructuras que son los diagramas entidad / Relación (DER).

• Entidades

Una entidad es un objeto real o abstracto de interés en una organización que acerca del cual se puede y se quiere guardar información . Su representación gráfica es:

Asociado al concepto de entidad surge el de Ocurrencia de entidad que es una realización concreta de la misma.

• Atributos

Un atributo es una propiedad o característica asociada a una entidad y común a todas las ocurrencias de la misma. Su representación es:

Asociado al concepto de Atributo surge el concepto de dominio. Dominio es el conjunto de valores permitidos para un atributo.

Hay 2 tipos de Atributos:

- ◆ Atributo identificador: Distingue una ocurrencia de entidad del resto de ocurrencias.
- ◆ Atributo Descriptor: Caracteriza una ocurrencia pero no la distingue del resto de ocurrencias de la entidad.

• Relacionales

Una relación es una asociación entre entidades. Entre 2 entidades puede existir más de un tipo de relación. Su representación es:

Asociado al concepto de Relación surge el concepto de ocurrencia de relación que es la asociación concreta de ocurrencias de entidad de las diferentes entidades. Una relación se caracteriza por dos propiedades:

- ◆ Grado: Es el nº de entidades que participan en la relación. Cuando una relación se asocia consigo misma es de grado 1(reflexiva), dos entidades distintas es de grado 2 o binaria, asocia tres entidades es de grado 3 o ternaria, y si asocia más de tres entidades es de grado n.
- ◆ Cardinalidad: Es el tipo de asociación establecido entre las entidades.
- ◆ Representación Gráfica

Para poder hacer el DER debemos definir todavía una serie de aspectos:

- ◇ La pertenencia de un atributo a una entidad o relación se representa uniendo el símbolo del atributo al símbolo correspondiente mediante un arco.
- ◇ Para reflejar la asociación existente entre 2 o mas entidades mediante una relación se unen los símbolos de dichas entidades al símbolo de la relación correspondiente mediante arcos.
- ◇ Cardinalidades

– Cardinalidad 1:1 ; Cada ocurrencia de la entidad A está asociada con una única ocurrencia de la entidad B y viceversa.

– Cardinalidad 1:N ; Cada ocurrencia de la entidad A está relacionada con una o varias ocurrencias de la entidad B y cada ocurrencia de la entidad B está asociada como máximo con una ocurrencia de la Entidad A.

· Cardinalidad M:N ; Cada ocurrencia de la entidad A está relacionada con una o varias ocurrencias de la entidad B y viceversa.

Por tanto hay que indicar si la participación de las entidades en la relación es obligatoria, opcional, condicional. En el DER éste aspecto se representa:

- Cardinalidad mínima
- Cardinalidad máxima

Asociada a cada entidad , puede darse:

(0,1) ; (1,1) ; (0,n) ; (1,n)

- ◇ Claves primarias

Las entidades tienen un atributo o conjuntos de atributos cuyos valores identifican de forma única una ocurrencia de dicha entidad de entre todas las demás, reciben el nombre de clave.

La entidad puede tener dos conjuntos de atributos diferentes que cumplan las condiciones establecidas para ser considerados claves. Se habla de claves candidatas y es necesario determinar cual va a ser utilizada de forma prioritaria, surgiendo entonces los dos conceptos siguientes:

- Claves primaria: Clave candidata elegida como clave

- Clave alternativa o secundaria: Resto de claves candidatas
- Clave ajena o foránea: Atributo o conjuntos de atributos de una entidad que son clave primaria en otra entidad.

Se representa subrayando dichos atributos.

◊ Entidades Débiles

La Entidad Débil está directamente relacionado con la denominada dependencia de existencia, es decir, una entidad tiene dependencia de existencia de otra cuando sin la primera, la segunda carecería de sentido.

A las entidades que tienen dependencia de existencia se les denomina entidades débiles, a las entidades que no presentan este problema se les denomina entidades fuertes o regulares.

La Entidad con dependencia ID (identificación) está relacionado con el concepto de clave primaria, se produce cuando una entidad no es identificable por el valor de sus atributos pero si por su relación con otra entidad.

La Entidad Débil con restricción de existencia con dependencia ID, no tienen una clave primaria sino tan solo un descriptor discriminador y por tanto necesita obligatoriamente la clave primaria de la entidad fuerte para poder identificar de manera única sus ocurrencias de identidad. La clave primaria de la entidad fuerte con el mencionado descriptor discriminador.

La representación Gráfica:

Nunca puede haber una asociación de dos entidades débiles. En una relación con Cardinalidad N:M nunca habrá entidades débiles. La dependencia de existencia no implica una dependencia ID pero sí a la inversa pues una entidad que depende de otra por su clave no tendrá sentido sin la existencia de ésta última.

◊ Tipos de Atributos

Hay dos tipos diferentes de atributos:

- **Multiocurrentes o multivaluados:** Son aquellos que para una misma ocurrencia de entidad pueden tomar más de un valor a la vez.
- **Atributos compuestos:** Son aquellos que agrupan en sí mismos por afinidad o por forma de uso más de un atributo.
- Relaciones Exclusivas

Dos o más relaciones son exclusivas si cada ocurrencia de una entidad solo puede pertenecer a una de las relaciones y se representa con un arco.

- Otras Representaciones gráficas

3.1. BACHMAN

Se representan los atributos escribiéndolos dentro de los símbolos de las entidades o relaciones a las que pertenecen. Las relaciones se representan por hexágonos.

3.2. MARTÍN

La Cardinalidad se refleja por símbolos gráficos representados sobre el arco:

- La Cardinalidad 1 son dos líneas verticales juntas y pegadas a la entidad
 - La Cardinalidad n es una línea vertical y un vértice pegado a la entidad
 - La relación se mantiene el rombo pero encerrado en un rectángulo, si la Cardinalidad es n.
- Generalización y herencia en el modelo EER.

El modelo EER permite representar las jerarquías existentes entre las entidades en el mundo real.

La Generalización es un tipo de abstracción que identifica una relación jerárquica.

La relación que se establece entre un supertipo de entidad y sus subtipos se denomina *es_un_tipo_de* o *Es_un* y su representación gráfica es:

Todo ocurrencia de un subtipo es una ocurrencia del supertipo. Lo contrario no sucede, por tanto las Cardinalidades se dan siempre:

- En el supertipo (1,1)
- En los subtipos (0,1) o (1,1)

La herencia es una abstracción que todo atributo del supertipo pasa a ser atributo de todos los subtipos.

Un tipo de entidad puede ser un subtipo para más de un tipo de entidad con lo que puede mantener diferentes relaciones jerárquicas dando lugar a lo que se denomina herencia múltiple. La relación puede ser parcial o total y con solapamiento o sin solapamiento.

- Pasos a seguir para obtener un modelo conceptual

Según BATIN para construir un esquema E-R destacan 4 estrategias:

1ª Descendente: Se refinan los conceptos de forma progresiva. Partiendo de una única entidad que describe el sistema se va descomponiendo sucesivamente con mayor nivel de detalle. Su desventaja principal es que requiere un alto nivel de abstracción.

2ª Ascendente: Se construyen los conceptos a partir de otros elementales.

Ventaja: Facilita las decisiones de diseño

Inconvenientes: Necesidad de reestructuración

3ª Incide_out: Es el más usual. Empieza creando un esquema E-R completándose a medida que se examina el esquema percibido en lenguaje natural y el resto de entidades y relaciones hasta ocupar todo el papel.

Ventaja: Facilita describir nuevos conceptos relacionados

Inconvenientes: Sólo se tiene una visión global del sistema al final

4º Mixto: Es una mezcla de los anteriores.

· Relación entre DFD y DER

–¿Q se construye primero el DFD o el DER?: Normalmente se desarrollan a la vez

–¿Es necesario construir los dos modelos: DFD y DER?: La mayoría de los sistemas actuales son lo suficientemente complejos en cuanto a funciones y datos como para hacer aconsejable la realización de ambos modelos.

· Relaciones de grado mayor que dos

Cuando se presentan relaciones de grado mayor que dos hay que analizar si la relación es realmente de tal grado o si puede descomponerse en otras de menor grado. Las relaciones de grado tres tienen la Cardinalidad de una de las entidades e1 con respecto a las otras dos (e2,e3) es el nº máximo y mínimo de ocurrencias de e1 que están asociados con uno de e2 y e3 ya asociados en la relación.

· Control de redundancias

Un elemento es redundante cuando puede ser eliminado sin pérdida de semántica. Existen 2 formas :

- Atributo: Es redundante si se deriva de cálculos realizados con otros atributos.
- Relaciones: Es redundante si está formada por una combinación de otras existentes, si podemos realizar la misma asociación de entidades por medio de otras relaciones existentes en el modelo.

TEMA 6

· SISTEMAS GESTORES DE BASES DE DATOS (S.G.B.D)

· Introducción

· Concepto de base de datos

Los sistemas orientados al proceso son datos q se almacenan en ficheros diseñados para una determinada aplicación; las aplicaciones se implantan independientes unas de otras y no comparten datos. Puede que se de una redundancia de datos al duplicar información, lo que conlleva:

- Una ocupación inútil de memoria
- Un aumento de tiempos de procesos al repetir los mismos controles para los datos
- La posibilidad de una inconsistencia de datos, es un problema grave pero su solución consiste en la integración de datos, surgiendo sistemas de bases de datos en los cuales los datos se estructuran y

mantienen en un conjunto organizado que no está diseñado para una aplicación concreta, sino que tiende a satisfacer las necesidades de información de toda la empresa, así surgen los sistemas orientados a los datos.

Una base de datos es un conjunto no redundante de elementos interrelacionados entre sí que se pueden procesar simultáneamente por varias aplicaciones.

· Ventajas e Inconvenientes de las bases de datos

Las ventajas que tienen los sistemas de bases de datos son:

- ◆ Independencia de los datos respecto a los tratamientos, un cambio en los datos, no altera a los programas ni viceversa.
- ◆ Coherencia de datos y resultados ya que los datos se recogen, almacenan y modifican una sola vez.
- ◆ Mejor disponibilidad de los datos para el conjunto de los usuarios que pueden compartir datos.
- ◆ Mayor flexibilidad para atender demandas cambiantes de los usuarios
- ◆ Mayor valor informativo ya que puede haber datos cruzados y ya que los distintos elementos están interrelacionados.
- ◆ Mejor documentación de la información que antes. En las bases de datos parte de la documentación está integrada con los datos.
- ◆ Reducción del espacio de almacenamiento

Los inconvenientes que tienen son:

- ◆ Instalación del equipo físico y lógico
- ◆ Necesidad de personal especializado
- ◆ Implantación larga y difícil con falta de rentabilidad a corto plazo
- ◆ Desfase entre teoría y práctica ya que todavía no son realidad algunos aspectos teóricos en cuanto a prestaciones.
- ◆ Niveles de abstracción en una base de datos (b.d.)

Los sistemas informáticos pueden ser observados de dos formas:

1ª– Lógica: Vista que tiene el usuario

2ª–Física: Es la forma real en que los datos están almacenados.

Aparece un nuevo nivel de abstracción intermedio siendo 3 niveles de abstracción:

1º– Estructura Lógica del usuario; denominada vista o esquema externo, se pueden interpretar como la visión que tiene cada usuario en particular de la base de datos; en cada vista sólo se deben encontrar los datos; En cada vista sólo se deben encontrar los datos y relaciones que necesite cada

usuario así como las restricciones de uso.

2º–Estructura Lógica Global: denominada modelo conceptual o esquema , la visión del conjunto de datos de toda la empresa y sus relaciones. Es la visión que tiene el administrados o persona encargada de el control de la base de datos completa, e incluye las restricciones de integridad y confidencialidad.

3º–Estructura física: denominada modelo físico o esquema interno, forma en que se organizan los datos en el almacenamiento físico e incluye:

- ◊ Estrategias de almacenamiento:
 - Asignación de espacios de almacenamiento para los datos
 - Estrategia de emplazamiento utilizada para optimizar el tiempo y espacio en memoria secundaria
 - Tratamiento de los desbordamientos
- ◊ Caminos de accesos:
 - Especificación de claves (primarias y secundarias)
 - Especificación de índices o punteros
 - Especificación de claves de ordenación
- ◊ Otros aspectos como:
 - Técnicas de compresión de datos
 - Técnicas de criptografía (codificar, cifrar información)
 - Técnicas de ajuste o afinamiento a efectos de eficiencia(tuning)
 - Técnicas de optimización
 - Traslación o correspondencia del esquema interno al conceptual

3. El administrados de la base de datos (ABD)

El ABD es el responsable del diseño, control y administración de la base de datos y puede ser una sola o un conjunto de personas, entre sus funciones :

- Determinar qué información va a almacenar en al base de datos y su estructura lógica.
- Decidir la descripción física
- Definir las vistas o esquemas externos para los programas
- Definir autorizaciones de uso de las bases de datos
- Definir estrategias de mantenimiento y controlar el rendimiento y los aspectos de explotación y uso
- Mantener la base de datos según cambia los requisitos del usuario

Entre las herramientas que utiliza el administrados están:

- Las utilidades del SGBD
- Herramientas CASE
- El DD que reúne toda la información sobre los datos almacenados
- El Sistema de Gestión de la base de datos (SGBD)

4.1. Historia y Evolución

En el año 1938 Claude E. Shannon propone la utilización de las funciones AND,OR,NOT con n° binarios dando lugar a lo que se ha denominado álgebra de boole.

En 1889 aparecen las tarjetas perforadas

En 1890 Hollerith crea el equipo para procesar las tarjetas

En 1944 aparece una máquina llamada MARK I

En 1946 aparece el ENAC construida por J.P.Eckert y J.W.Mauchly, el primer ordenador electrónico.

En 1947 aparece otra máquina llamada EDSAC que incorpora memoria interna direccionable para obtener la multifuncionalidad y tambien incorpora el diseño de una arquitectura de ordenadores.

A mediados y finales de los años 40 aparecen los discos magnéticos lo que da lugar a investigaciones y hace que aparezca los primeros S.O. y como consecuencia aparecen la Gestión de ficheros.

En los años 50 aparecen los primeros desarrollos de software usando lenguajes de programación y almacenando los datos en ficheros.

A principios de los años 60 aparecen los primeros sistemas de Gestión de ficheros no dependientes del S.O. son los precursores de los SGBD actuales.

En 1963 aparece el concepto de base de datos.

A mediados de los años 60 aparece la 1ª Generación de SGBD que están basados en el modelo CODASYL. Algunas características del SGBD:

- ◆ Separan la descripción de datos de los programas de aplicación que los utilizan.
- ◆ Aportan los primeros lenguajes de acceso a los datos de tipo navegacional.

Los SGBD mas representativos son: – IDMS; sistema en red

–IMS; modelo jerárquico de IBM

En 1970 se da origen a la 2ª Generación de los SGBD.

4.2. Características

La base de datos es un depósito único de datos disponible para toda la organización y que los usuarios ven solo la parte de los mismos que les interesa. Esto es posible al SGBD que es un conjunto de programas, procedimientos, lenguajes, etc, que suministran tanto a los usuarios no informáticos como a los analistas, programadores o al administrados los medios necesarios para describir, recuperar y manipular los datos, manteniendo su integridad, confidencialidad y seguridad.

Las funciones básicas son:

- ◆ Descripción o definición: Se lleva acabo con un Lenguaje de Definición de datos (LDD/DDL) que permite definir la base de datos a los 3 niveles(físico, lógico, lógico global)
- ◆ Manipulación de datos: Permite a los usuarios buscar, añadir, borrar o modificar los datos a los que tengan acceso, acabo mediante un lenguaje de manipulación de datos (LMD)
- ◆ Utilización: Proporciona un

conjunto de procedimientos para el administrador y se realiza a través de un lenguaje de control de datos (LCD)

- Tipos

Se clasifican en 3 grupos:

- ◇ Jerárquicos: Los datos se representan siguiendo una estructura jerárquica en la que los mas importantes se concentran en niveles superiores y los menos en niveles inferiores, es un modelo muy difundido en grandes equipos aunque va siendo sustituido por el relacional.
- ◇ En Red: Es similar al jerárquico pero un segmento hijo puede tener más de un padre.
- ◇ Relacionales: Es la tendencia más actual, los datos y sus relaciones se representan en forma de tablas.
- ◇ El Diccionario de datos

El DD son herramientas que independientemente ofrecen el necesario complemento a sus funciones para ayudar al administrador de la b.d. en su tarea, puede ser considerado como una b.d. más del sistema.

Los DDs se definen como sistemas que contienen información relativa a los datos existentes en la b.d. sus relaciones y su utilización.

El DD contendrá:

- Toda la información necesaria para definir una entidad o conjunto de datos
- Las descripciones de los diferentes esquemas que componen la b.d.
- Los procedimientos de autorización y validación de accesos
- Las referencias relativas a los programas y usuarios que utilizan la

b.d.
• Interacción
del
usuario
con
el
SGBD

En
general
los
usuarios
informáticos
y
entre
ellos
el
administrador
de
la
b.d.
necesitan
medios
potentes
y
flexibles
para
definir,
extraer,
y
manipular
datos
de
la
b.d.
en
el
lenguaje
de
programación
que
están
habituados
a
manejar
(lenguaje
anfitrión).
Al
conjunto
de
sentencias
de

manipulación
del
SGBD
que
permiten
el
acceso
a la
b.d.
se
le
llama
lenguaje
huésped.

El
usuario
final
necesitará
comunicarse
con
la
b.d.
de
una
manera
sencilla,
esto
puede
conseguirse
bien
por
un
lenguaje
de
manipulación
que
tenga
una
sintaxis
sencilla
o
bien
por
medio
de
tratamientos
para
metrizados
que
suelen
presentarse

al
usuario
en
forma
de
menús

- Interrelación
del
SGBD
con
el
S.O.

El
funcionamie
del
SGBD
está
relacionado
con
el
S.O.,
la
diferencia
principal
entre
el
modo
de
acceso
a un
fichero
y a
una
b.d.
es:

Cuando
se
trata
de
una
base
de
datos
el
programa
de
aplicación
se
dirige
al
SGBD
el
cual
a
través
del
S.O.
accede
a la
b.d.

TEMA
7

DISEÑO
ESTRUCTU
DE
SISTEMAS:

- Introducción

A

partir
de
los
DFD's
y
del
MER
en
el
DISEÑO
se
consigue:

1º
En
los
procesos,
conseguimos
una
arquitectura
detallada
de
prog.
y
mod.

2º
En
los
datos,
tendremos
un
modelo
físico
y
otro
lógico.

En
la
IMPLEMEN
se
consigue:
1º
En
los
datos
un
esquema
de
b.d.
y de

ficheros.
/ 2º
En
los
procesos
un
cuaderno
de
carga.

Con
ellos
se
hace
la
codificación
y
programación
del
sistema.

2.Diseño
detallado
orientado
al
flujo
de
datos:

Establece
la
transición
del
DFD
a
una
descripción
de
la
estructura
del
programa,
representada
por
un
diagrama
de
estructuras.

2.1.
Conceptos:

El
SW
se
divide
en
elementos
(módulos)
que
se
integran
entre
si
para
satisfacer
los
requisitos
del
sistema.
Así
se
facilita
el
manejo
del
SW.

MÓDULO:
Unidad(conj
de
sentencias)cl
definida
y
manejable
con
interfaces
perfectamen
de
finidas.
Cada
uno
recibe
un
nombre
significativo
El
sist.
Esta
compuesto
de
mod.
Organizados
Mods.del

sist.=

2.2
Diagrama
de
estructura:

muestra
la
modularizac
del
sist.
(división
del
sist.
en
mods.)Incluy
jerarquía,
org.
y
comunic.
entre
mods.
Sus
elementos
son:

–*Módulos*
–Paso
de
datos:

–*Conexione*
entre
módulos:
–Estructura
alternativa:

–Estructura
repetitiva:

–Módulo
predefinido:
–Paso
de
datos
de
control:

Diferencias
entre
diagrama

de
estructuras
y
organigrama

1ª
Un
org.
muestra
el
flujo
de
control,
o
secuencia
de
pasos
a
ejecutar.

2ª
Un
diag.
de
estruc.
Muestra
la
jerarquía(sub
de
funciones)
y
las
llamadas
q se
hacen
de
arriba
abajo
y de
izq.
a
der.

2.3
Tabla
de
interfaz:

Representa
los
parámetros
que

se
pasan
entre
módulos,
sirve
de
apoyo
al
diag.
de
estruc.
cuando
este
tiene
muchos
parámetros.
Muestra
para
cada
llamada:
El
módulo
llamado
y
cada
parámetro
formal(E/S,
uso,
significado).
Para
el
uso
se
usan
abreviat.
(procesado,
modific,
transferido
a
otro
mod
sin
modif.,
v.de
control,
tranferido
con
modif.).

3.
Estrategias
de

diseño:
(La
ERS
plantea
dos
diferentes).

3.1
Análisis
de
transformaci

Un
DFD
tiene
caract.
De
transformaci
cuando
los
datos
que
entran
van
caminando
hasta
llegar
a un
núcleo
donde
se
transforman
para
moverse
hasta
llegar
a la
S.
El
DFD
será
así:

M.P.
mod
princip.

M.C.E
m
controlador
de
E

M.C.T
m
cont.
de
transformaci

M.C.S
m
cont.
de
S

Estudiaremos
los
flujos
de
datos
y
sus
transform.
(E/S).
E:
Parte
de
los
procesos
que
elaboran
los
flujos
de
datos
de
E
hasta
que
estos
se
pueden
considerar
como
E
reales
al
sist.
Estos
datos
darán
lugar
a
las
S

del
sist.

Una
vez
aislados
el
centro
de
transformaci
y
sus
límites
con
el
flujo
de
E y
el
de
S,
las
transformaci
se
convierten
en
módulos
subordinado
Tb
hay
que
introducir
módulos
predefinidos
que
nos
den
las
E/S
del
sistema.

Después
refinaremos
la
estructura
del
sist.
Obtenida(rec
el
nivel
de

profund.
Del
diag.
De
estruc.
Después
a
ese
diag.
Se
le
añaden
los
parámetros
necesarios
para
cada
módulo.
Al
final
revisamos
el
diag.
De
estruc.,
observando
que
el
orden
de
ejecución
el
correcto.

3.2
Análisis
de
transacción:
(El
DFD
tiene
esta
forma).

Camino
de
transacción
1

Camino
de
transacción

Camino
de
transacción
3

Centro
de
transacción:
Es
el
centro
de
flujo
de
información
desde
el
que
nacen
muchos
caminos
de
acción(por
donde
pasa
la
inf.)alternati
y
excluyentes.
El
flujo
de
inf.
a lo
largo
de
un
camino
puede
ser
de
trans.
O
de
transac.
Para
hacer
el
análisis
de

transac.
Se
hace
casi
lo
mismo
que
para
el
otro
pero
en
la
conversión
del
DFD
en
la
estruc.
Del
sist.
Las
E
son
iguales
y a
las
S se
les
añade
un
mod.
Controlador
para
cada
camino
de
flujo
de
acción.

M.C.C:
mod.
Controlador
de
caminos.

4.
Calidad
del
diseño:

- Evaluación de la calidad de un diseño:

Para ver la calidad del SW la medimos con dos métricas: Acoplamiento y cohesión. Uno de los principios del diseño estructurado es que el sistema debe ser dividido en un conjunto de módulos q resulte manejables, pero se debe dividir dejando los módulos indep.(mín. acoplam.)

y
que
cada
mod.
lleve
a
cabo
una
función
relacionada
con
el
problema(ma-
Cohesión).

4.1
ACOPLAM.

Mide
el
grado
de
independencia
entre
los
mod.
Un
buen
diseño
debe
minimizar
el
acoplam.(au-
la
indep.)m,
por
las
sig.
Razones:

1º
Cuanto
menos
conexiones
existan
entre
2
mod.,
menos
oportunidad
hay
de q

aparezca
el
efecto
RIPPLE
(1defecto
de 1
mod.
aparece
afectando
a
otro).

2°
Se
desea
poder
cambiar
un
mod.
sin
q
afecte
a
otro.

3°
Mientras
se
mantiene
un
mod.
no
debemos
preocuparno
de
otro,
máxima
sencillez.

NIVELES
DE
ACOPLAM

—
Normal
//De
Datos//Stam
Control//Ext
Contenido.+

Niveles
deseables:

1
NORMAL:
2
mod.
están
acoplados
así
si
no
se
pasan
ningún
parámetro.

2
DE
DATOS:
Los
mod.
se
comunican
con
parámetros,
siendo
cada
uno
una
unidad
elemental
de
datos
y
no
un
elemento
de
control.

3
STAMP:
Si
hacen
referencia
a la
misma
estructura
de
datos,
no
siendo
esta
una

Estr.
De
datos
global,
lo q
implica
que
el
nombre
de
la
Estr.
es
pasado
a
través
del
prog.
Como
parámetro.

4
DE
CONTROL
: Si
entre
ellos
se
pasan
como
parámetros
elem
de
control,...

Niveles
no
deseables:

5
EXTERNO:
Cuando
ambos
hacen
referencia
a
una
variable
local

6
COMÚN

O
GLOBAL:

Igual
q el
Ext.
Y
se
compart1
estruc
de
datos
global.

Z
POR
CONTENID

Cuando
un
mod.
hace
referencia
al
interior
de
otro,
este
acoplamiento
es
patológico
y
totalmente
inaceptable.

Si 2
mod.
están
acoplados
de
mas
de
una
manera
su
acoplam
se
define
por
el
peor
de
los
tipos

de
acoplam
que
tenga
presentes.

4.2
COHESIÓN

Mide
el
grado
de
integración
de
los
elem
de
un
mod.,
para
org
todos
los
elem
de
forma
que
los
q
estén
más
relacionados
a la
hora
de
realizar
una
tarea
pertenezcan
al
mismo
mod.
Y
los
elem
relacionados
entre
sí
en
mod.
Se

parados.
ELEMENTO
cualquier
pieza
del
mod.
Que
lleva
a
cabo
un
trabajo
o
define
un
elemento.
TIPOS
DE
COHESIÓN
DE
+A
–

Funcional/Se

1º
FUNCIONA
Todos
los
elem.
Del
mod.
Están
relacionados
en
el
desarrollo
de
una
única
función
identificable

2º
SECUENCI
Cuando
el
mod.
representa
el
emplazamien
físico

de
varios
mod.
Con
cohesión
funcional.
Cada
mod.
Tiene
sentencias
que
deben
ser
ejecutadas
según
una
secuencia
específica,
ya q
la S
de
una
es
la E
de
otra.

3º **COMUNICA**

Cuando
todos
los
elem
del
mod.
Operan
sobre
los
mismos
datos
de
E y
S.
Aquí
el
orden
de
ejecución
no
importa.

4º

PROCEDURAL

Cuando
los
elem
de
procesam
del
mod.
Están
relacionados
y
deben
ejecutarse
en
un
orden
específico.

**5°
TEMPORAL**

Cuando
el
mod.
Tiene
tareas
relac.
Por
el
hecho
de q
todas
deben
hacerse
en
el
mismo
intervalo
de
tiempo.

**6°
LÓGICA:**

Un
mod.
Que
realiza
tareas
relacionadas
lógicamente
es
así.

7º
CAUSAL:
Cuando
entre
los
elem
q
componen
el
mod.
No
hay
ninguna
relación
con
sentido.
Es
una
cohesión
muy
perjudicial.

4.3
CONCLUSIÓN

Cuanto
más
vinculados
estén
los
elem
de
un
mod.
Más
fuerte
es
su
cohesión,
lo q
lleva
a un
mejor
acoplamiento
y a
un
mejor
sist.(el
SW
es
fácil
de

mantener
y
los
mod.
Fáciles
de
usar).A
medida
que
aumenta
el
núm.
De
mod.
El
coste
por
interfaz
crece
y el
esfuerzo
para
desarrollar
cada
mod
decrece.

Coste
o
esfuerzo
Coste
total
de
SW

Región
de

Coste
min.

Coste
por
mod.

Nº
de
mod.

• **Introducción**

El

análisis
de
datos
es
un
análisis
estático
basado
en
la
información
y
sus
rela-ciones,
mientras
que
el
análisis
de
flujo
de
datos
es
dinámico
basado
en
los
movimientos
de
esa
información
a
través
de
los
procesos.

El
análisis
de
datos
conlleva
varias
actividades
desde
el
estudio
de
la
realidad
existente
hasta

la
implementación
física
de
la
base
de
datos
en
un
sistema
de
gestión
comercial
existente
en
el
mercado,
hay
varias
etapas:

- ◆ Etapa de Análisis y diseño conceptual. Se analiza la realidad hasta diseñar un esquema conceptual que muestre las características principales de los datos y sus relaciones. Se utiliza el modelo

entio
relac
◆ Etap
de
Dise
Lógi
A
parti
del
esqu
conc
se
obtie
un
esqu
lógic
de
base
de
dato
segú
un
mod
lógic
conc
Jerár
Rela
en
Red
◆ Etap
de
Dise
físic
A
parti
del
esqu
lógic
se
proc
a
la
impl
físic
en
un
SGE
com
conc

2.
**Diseño
Lógico**

**de
Datos.
El
modelo
relacional**

Es la etapa siguiente a la de análisis de datos y su objetivo es pasar del modelo conceptual a un modelo lógico en el que se mejoren las posibles redundancias que pueda tener el modelo conceptual. Para obtenerlo hay varios métodos según el tipo de base de datos q se

vaya
a
diseñar:
Jerárquica,
Relacional,
en
Red
y
nosotros
utilizaremos
el
modelo
relacional.

Este
modelo
es
posterior
a
los
otros
dos,
está
basado
en
una
teoría
matemática
del
cálculo
relacional
propuesta
por
Codd
y
esta
teoría
se
basa
en
estructurar
lógicamente
los
datos
en
tablas
o
relaciones
que
sean
independientes
del

modo
de
almacenamiento
En
1990
apareció
una
nueva
obra
suya
en
la
que
presenta
la
2ª
versión
del
modelo
relacional
en
la
que
propone
330
características
que
deben
tener
según
él
un
sistema
para
que
sea
considerado
relacional,
en
el
mercado
se
reconoce
como
tales
al
DB2,
ORACLE,
etc.

2.1. Estructura

**del
modelo
relacional**

Todo
modelo
lógico
de
datos
presenta
una
serie
de
propiedades
estáticas
q
son
las
reglas
de
generación
de
las
estructuras
lógicas
de
datos
y
las
restricciones
sobre
estas
estructuras
y
propiedades
dinámicas
que
comprenden
las
operaciones
autorizadas
sobre
dichas
estructuras
de
datos.

2.1.1.
Estática
del
modelo
relacional.

*

Relación
o
tabla:
Es
la
estructura
de
datos
utilizada
en
el
modelo
relacional
de
Codd,
se
puede
representar
como
una
matriz
de
datos
en
la
que
al
conjunto
de
columnas
se
le
denomina
atributos
y
vienen
identificados
por
un
nombre
y al
conjunto
de
filas
se
le
denominan
tuplas
(ocurrencias)

de
la
relación).

Como
notación
gráfica
podemos
utilizar:

- ◆ Representar en forma de tabla:
:
En la 1ª fila se pone el nombre de todos los atributos y en las filas restantes van cada una de las ocurrencias de la relación (tuplas).
- ◆ Representar en Form Normal Nom de la tabla segun

de
todo
los
atrib
entre
paré
subr
la
clav
prin

*

Grado
de
la
relación
(tabla):
Es
el
nº
de
atributos
o
columnas
de
la
misma

*Clave
candidata
:
Atributo
o
conjuntos
de
atributos
que
permiten
identificar
unívocamente
cada
ocurrencia
de
la
relación.

*Clave
candidata
no
redundante
:
Clave
formada

por
un
conjunto
de
atributos
que
si
se
elimina
alguno
de
ellos
deja
de
ser
clave
candidata.

*Clave
principal:
Una
de
las
claves
candidatas
no
redundantes

*Clave
ajena
o
extranjera:
Atributo
de
la
relación
que
es
clave
principal
de
otra
relación

*Valor
nulo:
Significa
información
desconocida
Un
nulo
no

es
un
valor,
se
trata
de
una
marca
que
indica
que
el
atributo
aún
no
tiene
valor
asignado.

*Restricción
de
integridad:
Las
restricciones
son
ocurrencias
no
permitidas,
estas
reglas
deben
aplicarse
a
toda
base
de
datos,
las
mas
importantes
son:

- ◆ Integridad de Entidad: Ninguna atributo que forme parte de la

clav
prin
puec
toma
valo
nulo
♦ Integ
refer

2.1.2.
Dinámica
del
modelo
relacional.
Operadores

Codd
propuso
una
serie
de
lenguajes
denominados
lenguajes
relacionales
que
tienen
como
característica
común
que
no
es
preciso
especificar
el
procedimien
a
seguir
para
obtener
la
información.

La
dinamica
del
modelo
relacional
consiste
en
una
serie

de
operadores
que
permiten
pasar
de
un
estado
a
otro
de
la
base
de
datos.
Estos
lenguajes
operan
sobre
un
conjunto
de
tuplas
y se
dividen
en 2
tipos:

- ◆ Álge
Rela
Ope
sobre
relac
y
los
resu
suele
ser
a
su
vez
relac
en
ellos
hay
que
espe
qué
oper
quer
aplic
para

obte
el
resu
♦ Cálco
Rela
En
estos
leng
sólo
es
prec
india
cuál
es
el
resu
que
se
quie
obte
expr
med
cálco
de
pred
,
es
decia
desc
qué
conc
debe
cum
las
tupla
para
ser
selec

2.1.2.1.
Álgebra
Relacional

Codd
demostró
que
cualquier
acceso
a
una
base
de
datos

relacional
podía
realizarse
utilizando
estos
operadores.
Extrajo
de
la
teoría
de
conjuntos
los
operadores
unión,
diferencia
y
producto
cartesiano
a
los
que
añadió
la
selección
o
restricción
y la
proyección,
estos
5
operadores
son
los
denominados
Operadores
Primitivos
Q
Fundamenta
existiendo
otros
derivados
o
adicionales
como
la
combinación
la
intersección,
la
agrupación,
la

división
que
pueden
obtenerse
a
partir
de
ellos.

2.1.2.1.1.
Operadores
fundamental

OPERADORES
UNARIOS:

*Selección
o
Restricción:
Se
denota
con
la
letra
(sigma)

R=
predicado
(A)

Esta
operación
devuelve
una
nueva
relación
R
con
las
tuplas
de
la
relación
A
que
cumplen
la
condición
especificada
en
el
predicado.

Se admiten los operadores de comparación ($\geq$, $\leq$, $>$, $<$, $=$) y los conectores (and, or, not)

*Proyección (π):
Su sintaxis es:

$R = \pi_{\text{atrib1, atrib2, ..., atribN}}(A)$

$S = \text{predicador}(B)$

La proyección de la relación A según los atributos dados es una relación R cuya cabecera y cuerpo estarán formados por

el
conjunto
de
todas
las
tuplas
de
A,
definidas
sobre
los
atributos
especificado
Devuelven
columnas.

OPERADOR BINARIOS:

*Producto
cartesiano:
Esta
operación
permite
combinar
información
de
cualesquiera
dos
relaciones,
si
existen
atributos
con
el
mismo
nombre
se
identifican
añadiéndole
el
nombre
de
la
tabla
separado
por
un
punto
a
cada
atributo.

Se
suele
representar
como
X

$R=A$
 $X B$

Aplicando
esta
operación
obtenemos
otra
relación
R
cuya
cabecera
está
formada
por
las
cabeceras
de
las
dos
tablas
y
tantas
tuplas
como
pares
ordenados
existan
por
cada
una.

*Unión
(U):
La
unión
de
dos
relaciones
A y
B
compatibles,
es
otra
relación
cuya

cabecera
es
idéntica
a la
de
A
(o a
B)
y
cuyo
cuerpo
está
formado
por
las
tuplas
perteneciente
a A
o a
B
eliminando
las
duplicadas.

*Diferencia
(-):
Es
una
relación
cuya
cabecera
es
idéntica
a la
de
A o
la
de
B y
cuyo
cuerpo
está
formado
por
las
tuplas
perteneciente
a A
pero
no a
B,
ésta

operación
por
tanto
permite
buscar
las
tuplas
que
están
en
una
relación
pero
no
en
la
otra.

*División
o
cociente
(:):
Sean
las
relaciones
R1
y
R2
cumpliendo
que
todos
los
atributos
de
R2
están
también
en
R1
y
definidos
en
el
mismo
dominio

R1
(A1,A2,...An
B1,B2,
...Bn)

R2

(B1,B2,Bn)

Si
se
consideran
los
dos
atributos
compuestos
A,
B
(A=A1,A2,A
B=B1,B2,Bn
la
división
de
R1:R2
es
otra
relación
con
la
cabecera
A y
con
cuerpo
formado
por
el
conjunto
de
todas
las
tuplas
de
R1
en
el
atributo
A
cuyos
valores
correspondie
en
el
atributo
B
incluyen
a
todos
los
valores

del
atributo
B
en
la
relación
R2.

OPERACIÓ
RENOMBR
Se
denota
con
la P
:

P r
(E)

E
puede
ser
una
relación
o
una
expresión
de
álgebra
relacional.

–Si
E es
una
relación
esta
operación
devuelve
la
misma
relación
pero
con
nombre
nuevo,
R.

–Si
E es
una
expresión
del

álgebra
relacional
la
operación
renombrar
devuelve
el
resultado
de
esa
expresión
pero
con
el
nombre
R.

Otra
forma
de
esta
operación
es
la
siguiente:

P r
(A1,A2,An)
(E)

E es
una
expresión
con
N
atributos
desde
A1
hasta
An.
Esta
operación
devuelve
el
resultado
de
la
expresión
E
con
el
nombre

R y
con
los
atributos
con
el
nombre
cambiado
a
A1,
A2,
An.

2.1.2.1.2.
Operaciones
adicionales

*Intersección
de
conjuntos:
La
intersección
de
dos
relaciones
compatibles
A y
B
es
otra
relación
cuya
cabecera
es
idéntica
a la
de
A o
B y
cuyo
cuerpo
está
formado
por
las
tuplas
que
son
comunes
a A
y B.
El

símbolo

es $\wedge$

.

$R = A - (A - B)$

$R = B - (B - A)$

*Reunión,

combinación

o

join:

La

reunión

de

dos

relaciones

R1

y

R2

no

necesariame

compatibles

pero

en

las

que

al

menos

existe

un

atributo

con

el

dominio

común,

sobre

un

predicado

P es

una

nueva

relación

R3

cuya

cabecera

está

formada

por

la

concatenación

de

los

atributos
de
 R_1
y
 R_2
y
cuyo
cuerpo
está
formado
por
las
tuplas
del
producto
cartesiano
de
 $R_1 \times R_2$
que
satisfacen
el
predicado
 P .
Dependiendo
del
tipo
de
Predicado
la
operación
reunión
recibe
distintos
nombres:

- ◆ Equi
El
pred
 P
es
una
expr
que
cont
el
oper
de
com
igua
($P=$)
- ◆ Thet
o

O-r
El
pred
P
es
una
expr
que
cont
un
oper
de
com
disti
a
la
igua
|O|

(sigma)
O (r
x s)

Sean
las
relaciones
 $r \in \mathcal{R}$
y s
(S)
y
sea
O
un
predicado
de
los
atributos
del
esquema
RUS

◆ Reun
natu
(
|X|
):
Es
una
equi
sobr
todo
los
atrib

com
exist
entre
las
relac
R1
y
R2.
Lo
prim
que
se
hace
es
el
prod
carte
de
dos
argu
desp
se
reali
una
selec
forz
la
igua
de
los
atrib
que
apar
en
amb
esqu
de
relac
y
final
elim
los
atrib
dupl

$r \mid x$
 $s =$
 π
 $RUS($
 σ
 $r.a1=s.a1$
 $\wedge$
 $r.a2=s.a2$

$\wedge \dots \wedge$
 $r.an=s.an$
 $(r \times$
 $s))$

Dadas
 2
relaciones
 $r \in \mathcal{R}$
 $y s$
 (S)
siendo
 R
 $S=\{a_1,a_2,\dots,a$
la
reunión
natural
de r
 $y s$
denotada
por
 $r|x|s$
es
una
relación
del
esquema
RUS.

2.1.2.2
Cálculo
Relacional

La
diferencia
que
hay
entre
el
cálculo
y el
álgebra
es
que
en
el
álgebra
hay
que
especificar
que
operadores

se
tienen
que
aplicar
a
las
relaciones
para
obtener
el
resultado,
en
el
cálculo
solo
es
preciso
indicar
cuál
es
el
resultado
que
se
quiere
obtener.
Los
lenguajes
de
cálculo
relacional
pueden
ser
de
dos
tipos:

- ◆ Orie
a
la
tupla
Una
varia
se
inter
com
si
se
repr
las
tupla
de

una
relac
♦ Orie
al
dom
Una
varia
se
inter
com
si
se
repr
los
valo
de
un
dom

2.1.2.2.1.
Cálculo
relacional
de
tuplas

Se
expresa:
 $\{t \mid$
 $F(t)$
 $\}$

Se
lee:
El
conjunto
de
tuplas
 t
que
cumplen
el
predicado
 F
siendo
 t
una
variable
de
tupla
varía
sobre
una
relación

2.1.2.2.2.
Cálculo
Relacional
de
dominios

Es
equivalente
al
anterior
pero
sus
variables
varian
sobre
dominios
en
lugar
de
hacerlo
sobre
tuplas.

2.1.2.3.
Reglas
para
las
claves
ajenas

Al
diseñar
una
base
de
datos
además
de
definir
las
claves
ajenas
hay
que
especificar
como
debe
actuar
el
sistema

en
caso
de
que
se
produzcan
ciertas
operaciones
sobre
tuplas
de
la
relación
referenciada
Casos:

- ◆ Operaciones de restricción
restricción (rest) :
Si
existe una
clave
primaria
que
es
clave
ajena
solo
se
permite
su
borra
o
su
modificación
sino
existen
tuplas
con
dicha
clave
en
la
relación
que
contiene
la
clave
ajena
- ◆ Operaciones de actualización

con
trans
en
casc
(cas
Si
el
valo
de
la
clav
prim
es
mod
auto
se
mod
el
valo
en
la
clav
ajen
y
si
se
borr
una
tupla
de
la
relac
que
cont
la
clav
prim
auto
se
borr
las
tupla
asoc
de
la
relac
que
cont
la
clav
ajen
◆ Ope

con
pues
a
nulo
(set
null)
Es
pare
a
la
ante
pero
el
borr
o
la
mod
de
tupla
de
la
relac
que
cont
la
clav
prim
refer
lleva
cons
pone
a
nulo
los
valo
de
las
clav
ajen
por
tanto
al
borr
o
mod
tupla
se
pier
la
conc
◆ Ope
con

pues
a
valo
por
defe
(set
defa
Es
igua
que
la
ante
pero
en
luga
de
pone
el
valo
nulo
a
la
clav
ajen
se
le
pone
el
valo
que
tiene
asoc
por
defe
este
valo
se
defin
al
crea
la
tabla
corre
◆ Ope
que
dese
un
proc
de
usua
El
borr

o
la
mod
de
tupla
de
la
tabla
refer
pone
en
marc
un
proc
defin
por
el
usus

• **Paso
del
Modelo
Conceptual
al
Relacional**

El
objetivo
del
diseño
lógico
es
conseguir
a
partir
del
modelo
conceptual
un
modelo
lógico
de
datos
formado
por
un
conjunto
de
relaciones
o
tablas
que
sean

fácil
de
usar
independiente
de
las
estructuras
lógicas
y
físicas
de
los
datos
Las
etapas
a
seguir
serán:

- ◆ Transformar los
objetos
conceptuales
en
relaciones
- ◆ Normalizar las
relaciones
obtenidas

El
diagrama
E-R
puede
ser
transformado
a un
conjunto
de
tablas
que
serían
directamente
implantables
en
cualquier
SGBD
Relacional
comercial. Se
las
siguientes
reglas:

- ♦ Toda entidad se transcribe en una tabla siguiendo las normas que indican según sea el tipo de entidad (fuente o debida)
- ♦ Para las relaciones entre sus grados y clases de pertenencia de las entidades parti en la misma

Entidades fuertes
o reguladas:

Cada entidad fuerte es representada por una tabla que

tomará
por
nombre
el
identificador
de
la
entidad
y
cuyas
columnas
se
corresponden
unívocamente
con
cada
uno
de
los
atributos
de
la
entidad,
tomando
como
nombre
los
identificadores
de
estos.

Entidades débiles:

Se
le
añade
a la
tabla
como
columnas
propias
los
atributos
que
formen
la
clave
primaria
de
la
entidad

fuerte
de
la
que
depende
la
entidad
débil.

Relaciones:

Ésta
tomará
por
nombre
el
identificador
de
la
relación
y
sus
columnas
se
corresponden
con
cada
uno
de
los
atributos
de
la
relación
tomando
como
nombre
los
identificadores
de
estos
además
a
las
tablas
de

las relaciones se les añade como atributos los claves primarias de las entidades que asocian las cuales en su conjunto formarán la clave primaria de la tabla de la relación.

3.1. Estudio del grado y las Cardinalidad

- ◆ Relación bina de grad 1:1

Existen dos posibilidades

1ª—

Se
obtendrá
una
tabla
que
tendrá
como
atributos
la
clave
principal
de
la
primera
entidad,
el
resto
de
sus
atributos,
la
clave
primaria
de
la
segunda
entidad,
el
resto
de
sus
atributos
y
los
atributos
de
la
relación.

R1(E1,atrs
E1,
E2,
atrs
E2,
atrs
A)

2ª–
Se
hacen
dos
tablas

correspondie
a
las
entidades,
seleccionand
una
de
las
claves
primarias
como
clave
ajena
en
la
otra
entidad,
esto
se
conoce
como
propagación
de
la
clave.

R1
(E1,atrs
E1)

R2
(E2,atrs
E2,E1,atrs
A)
/*propagació
de
la
clave
R1*/

Se
propaga
la
clave
de
la
entidad
con
Cardinalidad

1:1
a la
tabla
de
la
otra
entidad
o se
pasa
la
clave
primaria
de
la
entidad
opcional
a la
otra
como
clave
ajena.

R1
(E1,
E1,
E2,
atrs
A)

R2
(E2,
atrs
E2)

Se
expresa:
R1
(E1,
atrs
E1)

R2
(E2,
atrs
E2)

R3
(E1,
E2,

atrs
A)

◆ Rela
Bina
1:M
entre
dos
entid
regu

Existen
dos
posibilidades

1ª–
Se
hacen
las
tablas
de
cada
entidad
y la
clave
primaria
de
cada
entidad
con
Cardinalidad
M
aparecerá
como
clave
ajena
en
la
tabla
de
la
otra
entidad,
y
los
atributos
de
la
asociación

R1
(E1,
atrs

E1)

R2
(E2,
atrs
E2,
E1,
atrs
A)

2ª-Se
hacen
tres
tablas
pero
la
tabla
corresponde
a la
relación
tendrá
como
clave
primaria
de
la
entidad
que
se
encuentra
en
el
lado
M
de
la
relación
esta
solución
se
hace
cuando
la
relación
tiene
atributos.

R1
(E1,
atrs
E1)

R2
(E2,
atrs
E2)

R3
(E1,
E2,
atrs
A)

- ◆ Relación Bina-
1:M
entre
una
entidad
regu-
y
otra
entidad
débi-

Se
obtendrán
dos
relaciones
unos
con
los
mismos
atributos
de
la
primera
entidad
y
otra
con
la
clave
principal
de
la
entidad
regular
más
los
atributos
de
la
segunda
entidad

añadiéndoles
los
atributos
de
la
relación.
Sea
E2
la
entidad
débil.

R1(E1,
atrs
E1)

R2(E1,
E2,
atrs
E2,
atrs
A)

◆ Rela
Bina
de
grad
M:N

Se
obtendrán
las
tablas
correspondie
a
las
entidades
y a
la
relación.

R1
(E1,
atrs
E1)

R2
(E2,atrs
E2)

R3
(E1,
E2,

atrs
A)

–
Relaciones
n-arias
n>2

Se
obtendrá
una
relación
por
cada
entidad
participante
con
los
atributos
de
la
misma
más
otra
con
los
atributos
de
la
asociación
más
las
claves
primarias
de
todas
las
entidades
asociadas.
Supongamos
una
relación
ternaria:

R1
(E1,
atrs
E1)

R2
(E2,
atrs

E2)

R3

(E3,

atrs

E3)

R4

(E1,

E2,

E3,

atrs

A)

3.2.

JERARQUÍA

Para

transformar

tipos

y

subtipos

en

el

modelo

relacional

se

crea

una

tabla

para

la

entidad

genérica

o

supertipo

con

todos

los

atributos

comunes

y

una

tabla

para

cada

entidad

no

genérica

o

subtipo

que

tendrá
como
clave
principal
la
de
la
entidad
genérica
y
como
atributos
sus
atributos
específicos.

OPERACIÓN
ADICIONAL
DEL
ÁLGEBRA
RELACIONAL

División
o
cociente:

Sean
 R_1
y
 R_2
cumpliendo
que
todos
los
atributos
de
 R_2
están
en
 R_1 y
definidos
por
el
mismo
dominio.

$R_1(A_1, A_2, \dots, B_n)$
 $A = A_1, A_2, A_3, \dots, A_n$

$R_2(B_1, B_2, B_3, \dots, B_m)$
 $B = B_1, B_2, B_3, \dots, B_m$

Si
se
consideran
dos
atributos
compuestos
A y
B,
la
división
R1/R2
es
otra
relación
con
la
cabecera
A y
con
cuerpo
formado
por
el
conjunto
de
todas
las
tuplas
de
R1
en
el
atrib
A,
cuyos
valores
correspondie
en
el
atrib
B
incluyen
a
todos
los
valores
del
atrib
B
en
la
relación

R2.

Dadas
2
relaciones
 $a(A)$
y
 $b(B)$

$a:b=pi$
 $A-B(a)-piA$

• NORMALIZ

Su
objetivo
es
conseguir
un
conjunto
de
relaciones
óptimo
que,
una
vez
implementado,
físicamente,
forme
una
estructura
flexible
y
fácil
de
mantener
y
evite
anomalías
en
operaciones
de
manipulación
de
datos,
evite
redundancias
e
inconsistencias.

La
normalización

se
basa
en
la
sustitución
de
un
esquema
inicial
de
relación
por
un
conjunto
de
subesquemas
equivalentes
tales
que
eliminan
una
serie
de
anomalías
que
presenta
el
1º.
Estas
anomalías
son:

DE
INSERCIÓN
Imposibilidad
de
introducir
unos
datos
por
carecer
de
otros.

DE
ACTUALIZACIÓN
Necesidad
de
propagar
actualizaciones
por
tener

un
diseño
redundante.

DE
BORRADO
Pérdida
de
información
no
necesaria
que
conlleve
la
perdida
de
información
necesaria.

4.1
DEFINICION

DEPENDENCIA
FUNCIONAL
Se
dice
que
un
atributo
B
depende
funcionalmente
de
otro
A
de
la
misma
relación
si
cada
valor
de
A
tiene
asociado
en
todo
momento
un
valor
de

B,
se
representa
así:

$a \rightarrow b$
(b
depende
funcionalme
de
a)

$a \rightarrow b$

O
se
puede
representar
así:
a es
implicante
y b
el
implicado

Siempre
se
cumple
que
todos
los
atributos
de
una
relación
dependen
funcionalme
de
las
claves
candidatas
de
la
relación.

Las
DF
siempre
se
estudian
en
el

ámbito
de
una
relación,
es
decir,
no
se
estudian
las
relaciones
existentes
entre
atributos
de
las
distintas
relaciones.
Estas
dependencia
representan
siempre
relaciones
de
muchos
a
uno.

EQUIVALE
Se
dice
que
 2^A
atrib
A y
B
son
equivalentes
entre
sí si
B
depende
funcionalme
de
A y
viceversa.

Si a
b y
b a
entonces
a b

DEPENDEN
FUNCIONA
COMPLETA
Se
dice
que
un
atributo
B
tiene
DFC
de
un
grupo
de
atributos
A
(a1,a2...)
de
la
misma
relación.
Si
B
depende
funcionalme
de
A
pero
no
de
ningún
subconjunto
de
los
atributos
que
lo
forman.

a b ,
a1 /
b ,
a2 /
b .

Si x
es
simple(form
por
un
único

atributo)se
cumple
que
toda
dependencia
funcional
sobre
x es
completa.

DEPENDEN
TRANSITIV

A,
B y
C,
tres
atributos
o
grupo
de
atributos
de
una
relación
.Si
B
depende
funcionalme
de
A y
C
depende
funcionalme
de
B,
pero
A
no
depende
funcionalme
de
B,
se
dice
que
C
depende
transitivame
de
A, a
través
de

B.
Se
representa:

a b

/ a

– c

c

4.2

1º

FORMA
NORMAL:

Una
relación
R
está
en
1ºFN
si y
solo
si
todos
los
dominios
simples
y
subyacentes
contienen
solo
valores
atómicos.
La
intersección
fila-columna
contiene
un
valor.

4.3

2º

FORMA
NORMAL:

Una
relación
R
está
en
2ºFN
si

está
en
1ºFN
y
además
todos
los
atributos
no
principales
tienen
una
dependencia
funcional
completa
respecto
de
alguna
de
las
claves
candidatas.
Así
se
eliminan
las
dependencia
funcionales
no
completas
Para
pasar
la
relación
a
2ºFN
se
descompone
en 2
nuevas:

$R_1(A,D)$
siendo
la
clave
primaria
 $A \rightrightarrows R_2(A,E)$
la
clave
principal
es
 (A,B)

4.4
3º
FORMA
NORMAL:
Una
relación
R
está
en
3ºFN
si
está
en
2ºFN
y
no
existe
ningún
atributo
no
principal
que
dependa
transitivamente
de
alguna
clave.
Para
pasar
una
relación
a
3ºFN
se
descompone
en 2
nuevas
relaciones
sin
pérdida
de
información,
de
manera
que,
al
menos,
1 de
ellas
esté
en
3ºFN.

R1(B,C)
siendo
Pk
B

R2(A,B)
siendo
Pk
A

Para
que
una
relación
esté
en
3ºFN
sus
campos
deben
ser
mutuamente
dependientes
y
completamen
dependientes
de
la o
las
claves
candidatas,
es
decir,
cada
uno
de
sus
atributos
debe
depender
solamente
de
la
clave
y
no
de
otros
atributos.

**El
paradigma**

**orientado
a
objetos**

El
enfoque
orientado
a
objetos
proporciona
las
siguientes
ventajas:

Reutilización
de
los
componentes
software

Menores
efectos
colaterales

El
paradigma
orientado
a
objetos
encierra
una
visión
completa
de
la
ingeniería
de
software,
un
simple
uso
de
la
programación
estructurada
a
objetos
no
brindará
los
mejores
resultados,

los
ingenieros
del
software
deben
considerar
los
siguientes
elementos:

Análisis
orientado
a
objetos
(AOO)

Diseño
orientado
a
objetos
(DOO)

Programación
Orientada
a
objetos
(POO)

SGBD
Orientado
a
objetos
(SGBDOO)

El
proceso
Orientado
a
objetos
se
mueve
a
través
de
una
espiral
evolutiva
que:

1°.
Empieza
la

comunicación
del
usuario
donde
se
define
el
dominio
del
problema
y se
identifican
las
clases
básicas
del
problema
.

2º–
La
planificación
y el
análisis
de
riesgos
establecen
una
base
para
el
plan
del
proyecto
orientado
a
objetos.

3º–
El
trabajo
técnico
asociado
con
la
ingeniería
del
software
orientado
a
objetos
sigue

el
siguiente
camino
iterativo:

Identificar
las
clases
candidatas

Buscar
clases
en
biblioteca

Extraer
nuevas
clases
si
existen

Desarrollar
las
clases
si
no
existen
(AOO,DOO,

Añadir
las
nuevas
clases
a la
biblioteca

Construir
la
enésima
iteración
del
sistema

**Conceptos
de
orientación
a
objetos**

- **Clases
y
Objetos**

Los sistemas orientados a objetos integran 2 aspectos que los sistemas de información analizaban de forma separada:

Datos

Procesos

Un **sistema** se concibe como un conjunto de objetos que se comunican entre sí mediante mensajes.

Un **objeto** es una entidad percibida en el sistema que se está desarrollando.

se describe por sus propiedades o atributos y por los servicios y operaciones tambien llamados métodos que pueden proporcionar

El **estado de un objeto** viene determinado por los valores que toman sus atributos, valores que siempre han de cumplir las restricciones impuestas sobre ellos.

Los objetos se distinguen unos de

otros
por:

- ◆ Posee un estacionamiento que se puede cambiar.
- ◆ Tiene una identificación única.
- ◆ Posee un computador propio.
- ◆ Soporta interacción con otros objetos.
- ◆ Puede emitir y recibir mensajes.

Una **clase** es la implementación de un tipo de objeto, es decir, es una plantilla o patrón que describe una colección de objetos similares.

La
clase
encapsula
los
datos
y
los
procesos
que
manipulan
esos
datos.
Esto
posibilita
el
ocultamiento
de
información
y
reluce
el
impacto
de
efectos
colaterales
asociados
a
cambios.

El
nº
de
propiedades
de
un
objeto
debe
ser
el
mínimo
necesario
para
realizar
todas
las
operaciones
que
requiere
la
aplicación.

La

única
forma
de
alcanzar
los
atributos
es ir
a
través
de
alguno
de
los
métodos
que
se
proporcionar

Todos
los
objetos
de
una
clase
heredan
sus
atributos
y
las
operaciones
disponibles
para
la
manipulación
de
dichos
atributos.

Los
objetos
se
comunican
unos
con
otros
mediante
estímulos
y
mensajes,
un
mensaje
estimula

la
ocurrencia
de
cierto
comportamiento
en
el
objeto
receptor.
Una
operación
dentro
de
un
objeto
emisor
genera
un
mensaje
que
se
expresa:

[
DESTINO
,
OPERACIÓN
PARÁMETROS
]

Destino:
objeto
receptor

Operación:
Método
que
recibe
el
mensaje

Parámetros
Información
requerida
para
el
éxito
de
la
operación

Las

interacciones
entre
objetos
pueden
ser:

Estáticas:

Generalización:

Los
tipos
o
las
clases
se
organizan
en
jerarquías
de
supertipos
y
subtipos
que
presentan
herencia
y
que
se
corresponden
con
el
concepto
es
un.

Agregación:

Permite
construir
objetos
compuestos
y se
corresponden
con
el
concepto
parte
de.

Dinámicas:

Consisten
en
mensajes

por
medio
de
los
cuales
los
objetos
solicitan
la
prestación
de
servicios
y
entrega
si
proceden
los
resultados
que
se
obtienen
cuando
se
lleva
acabo
un
cierto
servicio.
Proporcionar
importantes
beneficios:

Los
detalles
de
implementación
interna
de
los
datos
y
procedimientos
están
ocultos
al
mundo
exterior,
esto
reduce
la
propagación
de

efectos
colaterales
cuando
ocurren
cambios
.

Las
estructuras
de
datos
y
las
operaciones
que
las
manipulan
están
mezcladas
en
una
unidad
sencilla,
la
clase,
esto
facilita
la
reutilización
de
componentes

Las
interfaces
entre
objetos
encapsulados
están
simplificadas

• **Herencia**

Consiste
en
la
propagación
de
los
atributos
y
operaciones
de

un
objeto
através
de
la
jerarquía
de
clases;
cualquier
objeto
hijo
hereda
todos
los
atributos
y
operaciones
de
sus
padres.

Una
clase
puede
heredar
de
varias,
esto
se
llama
herencia
múltiple.En
general
este
tipo
de
herencia
complica
la
jerarquía
de
clases
y
puede
crear
problemas.

• **Polimorfismo**

Capacidad
de
que

un
mensaje
sea
interpretado
de
maneras
distintas
según
el
objeto
que
lo
recibe.
El
polimorfismo
indica
la
posibilidad
de
definir
varias
operaciones
con
el
mismo
nombre
diferenciand
unicamente
en
sus
parámetros.
Es
usual
que
la
vinculación
entre
el
nombre
de
un
servicio
y su
implementación
se
establezca
en
tiempo
de
ejecución
aunque
la

determinación
de
tipos
se
realice
en
tiempo
de
compliación
a
fín
de
evitar
errores
sintácticos
en
tiempo
de
ejecución.

El
seleccional
el
servicio
a
ejecutar
en
tiempo
de
ejecución
facilita
el
mantenimien
de
los
programas.

**El
desarrollo
orientado
al
objeto**

Se
distinguen
3
fases
del
análisis
estructurado

Análisis

(entender el dominio del problema)

Diseño (espacio de solución: como)

Implementación (se adapta la solución a un entorno específico)

Entre las metodologías utilizadas para el desarrollo orientado al objeto podemos distinguir entre dos grandes corrientes:

Metodología dirigida por los datos; se basan principalmente en la parte estructural de los

objetos
y
son
una
extensión
del
modelado
conceptual
en
el
MER.

Metodología:
dirigidas
por
las
responsabilidades
se
centran
en
las
acciones
que
puede
llevar
acabo
un
objeto.

Podemos
distinguir
los
siguientes
pasos:

Identificar
las
clases
con
sus
servicios
y
atributos

Establecer
las
interacciones
entre
las
clases,
estudiando
la

generalización
agregación
y la
colaboración

Añadir
las
clases
de
interfaz,
las
de
aplicación
y
las
claves
base.

Implementar
las
clases.

**Técnicas
de
desarrollo
orientado
al
objeto**

**• Tarjetas
de
clase**

Consiste
en
elaborar
por
cada
clase
una
tarjeta
o
ficha
en
al
que
se
apunta
sus
principales
características

Nombre

**Lista
de
superclases**

**Lista
de
subclases**

**Responsabil
de
un
objeto**
son
todos
los
servicios
que
proporcionar
para
todos
los
contratos
que
soportan.

Colaboraci
Representan
las
peticiones
por
parte
de
un
cliente
a
los
servidores
para
el
cumplimiento
de
las
responsabilid
de
dicho
cliente.

La
relación
Colaboracion

no
es
de
tipo
1:1
porque
cada
colaboración
atiende
a
una
responsabilidad
la
responsabilidad
puede
requerir
varias
colaboraciones

• **Formas
de
utilización
(uses
cases)**

Sus
principales
objetivos
son:

**Capturar
los
requisitos
funcionales
del
sistema,**
comunicar
requisitos
desde
el
punto
de
vista
del
usuario.

**Simplificar
la
construcción
de
los
modelos**

**de
objetivos.**
Un
modelo
de
formas
de
utilización
es
un
grafo
que
tiene
2
tipos:

Actor:
Representa
cualquier
elemento
de
intercambio
de
información
con
el
sistema.

**Forma
de
utilización:**
Una
secuencia
de
transacciones
en
un
diálogo
con
el
sistema
que
se
encuentran
relacionadas
por
su
comportamiento

Los
Actores
y

las
Formas
de
utilización
se
relacionan
a
través
de
arcos
de
comunicación

• **Modelo
de
Clases**

Todas
las
metodología
representan
los
conceptos
de
clase
y de
objetos
algunas
utilizan
2
diagramas
distintas
mientras
que
otras
sólo
emplean
uno
basado
en
extensiones
de
los
diagramas
de
E/R,
para
representar
las
clases,
es
un

rectángulo
en
cuyo
interior
figura
el
nombre
de
la
clase
y a
veces
tambien
los
atributos
y
los
servicios.

Hay
metodología
que
distinguen
3
categorías
de
clases:

Entidad:

Control:

Interfaz:

En
los
diagramas
de
clases
se
suelen
representar

4
tipos
distintos
de
asociaciones

1º–
Interrelacio
Con
sus

cardinalidad
igual
que
se
hace
en
los
DER
aunque
puede
cambiar
la
notación.

2º–
Generalización
Suele
representarse
con
un
triángulo,
media
circunferencia
o
una
flecha.
Algunos
representaciones
distinguen
entre
subtipos
solapados
o
exclusivos
y
entre
generaciones
totales
o
parciales.

3º–
Agregación
Se
suele
representar
con
un
triángulo
o
con
un

rombo

4º-

Comunicaci

Representa

como

un

objeto

de

una

clase

se

comunica

con

otro.

• **Otras técnicas**

Cada

metodología

introduce

nuevos

diagramas

para

expresar

mejor

alguna

parte

del

sistema

en

cualquiera

de

sus

vertientes,

algunos

de

estos

diagramas

son:

- ◆ Diag
de
trans
de
Esta
- ◆ Diag
de
even
- ◆ Diag
de

Inter
de
Obje
◆ Red
de
Petri
◆ Diag
de
Com
(igua
que
los
DFD
◆ Diag
de
Fluj
de
Obj
◆ Diag
de
Inter
o
plan
de
solu
de
probl
basa
en
la
meto
KAI
◆ Diag
de
Fluj
de
Acti
◆ Diag
de
Cuac
de
serv

**Metodología:
OMT**

La
metodología
OMT
propone
utilizar
la s
siguientes

técnicas:

Modelo de Objetos
(modelo de clases)

Modelo Dinámico:
Es un conjunto de diagrams de transición de estados que representa el comportamiento del sistema.

Modelo Funcional:
Corresponde a un DFD

Esta metodología parte de la idea de utilizar los mismos conceptos y notación a lo largo de todo el ciclo

de
vida,
construyendo
unos
modelos
a
los
que
se
van
añadiendo
más
detalles,
consta
de 5
fases:

Conceptualización:
Consiste
en
hacer
un
análisis
preliminar
que
identifica
el
problema
a
resolver,
la
técnica
utilizada
es
formas
de
utilización.

Análisis:
Describe
el
comportamiento
del
sistema
como
una
caja
negra
construyendo
los
3
modelos

anteriores.

Diseño del sistema:
Se toman las decisiones de alto nivel sobre la estructura global del sistema.

Diseño de Objetos:
Se refinan los modelos resultantes de la fase de análisis y se eligen los algoritmos de estructuras de datos pero con independencia de un lenguaje o entorno particular.

Implementa
Se

codifica
el
diseño
en
un
lenguaje
específico.

• **Fase
de
Análisis**

Se
obtiene
una
descripción
precisa
y
comprensible
desde
las
3
perspectivas
Estructural,
Dinámica
y
Funcional.

• **Modelo
de
Objetos:**

Establece
las
siguientes
etapas.

Identificar
objetos
y
clases

Preparar
un
DD

Identificar
asociaciones
y
agregaciones
entre
objetos

Identificar
atributos

Organizar
y
simplificar
las
clases
utilizando
la
herencia

Verificar
que
existen
caminos
de
accesos
para
las
consultas
más
probables

Iterar
y
refinar
el
modelo

Agrupar
clases
y
módulos

- **Módulo Dinámico:**
Para obtenerlo hay que seguir los siguientes pasos:

Preparar
escenarios
con
secuencias
de
interacción

típicas
entre
los
usuarios
y el
sistema

Identificar
eventos
entre
objetos

Preparar
una
traza
de
eventos
para
cada
escenario

Construir
un
diagrama
de
estados
para
toda
clase
de
objetos

Verificar
la
consistencia
de
los
eventos
asociados
a
los
objetos

• **Modelo
Funcional:**

Se
identifican
los
valores
de
E /

S

Se
construyen
DFD
para
mostrar
las
dependencia
entre
operaciones
y se
especifiquen
los
criterios
de
optimización

• **Diseño
del
sistema**

Se
especifica
la
arquitectura
del
sistema
que
se
divide
en
varias
etapas:

Organizar
el
sistema
en
subsistemas

Identificar
qué
objetos
pueden
actuar
y
cuales
no

Elegir
un

enfoque
para
la
gestión
de
almacenes
de
datos

Elegir
un
enfoque
para
la
implementación
del
control

Considerar
las
condiciones
del
entorno

Establecer
prioridades
entre
los
distintos
objetivos
del
diseño

• **Diseño
de
objetos**

En
esta
fase
se
lleva
acabo
la
estrategia
del
diseño
de
sistema
y se
completa
los

detalles:

Obtener
las
operaciones
para
las
clases
a
partir
de
los
otros
modelos

Diseñar
algoritmos
para
implementar
las
operaciones
definiendo
nuevas
clases
y
operaciones
si
fuese
necesario.

Optimizar
los
caminos
de
accesos
a
los
datos

Implementar
el
control
seleccionado
en
la
fase
anterior,identificar
los
caminos
principales
y
los

alternativos

Ajustar
la
estructura
de
clases
para
incrementar
la
herencia

Diseñar
las
interrelaciones

Empaquetar
las
clases
e
interrelaciones
en
módulos

Determinar
la
representación
exacta
de
los
objetos

**1.
Diseño
de
Procedimientos
de
Usuario**

Define
aquellas
tareas
que
deben
realizar
las
personas
para
que
funcione
el
sistema,

para
ello
habrá
que
indicar
que
tareas
tiene
que
realizar
cada
usuario
y
como
ha
de
efectuarlas.

Un
procedimien
de
usuario
debe
especificar
como
se
obtienen
los
datos,
el
medio,...

*
Los
procedimien
de
usuario
consta
de 2
partes:

1º–
Descripción
General:
Anota
la
finalidad
y la
descripcion
general
del
procedimien

2º–
Definir
como
se
van
a
realizar
estas
tareas.

A
este
fin
se
suelen
hacer
diagramas
de
flujo.

• **Diseño
de
la
Interfaz
de
usuario**

Define
como
interactúa
el
usuario
con
el
ordenador.
En
esta
fase
se
describirán
todo
lo
referente
a la
comunicación
del
sistema
con
el
exterior
(
diseño

de
pantallas,
...).

Se
han
utilizado
los
siguientes
tipos
de
interfaces
de
usuario:

- ◆ De **1965**
Pantallas
compartidas
con
menús
con
estructura
jerárquica
- ◆ De **1980**
Interfaces
gráficas
de
usuario
Ventanas
Iconos
Menús

Las
interfaces
presentan
varios
niveles:

Niveles
superiores
representan
menús

Niveles
inferiores
representan
pantallas
de
diálogo
o
funciones

concretas

El propósito de la interfaz es recoger de los usuarios la información del sistema y ponerla a la disposición de otros usuarios. La información recogida se llama **entrega del sistema** y se almacena en la b.d., la información suministrada se llama **salida del sistema**.

Las E /S pueden ser:

Interactivas

El usuario

se
comunica
directamente
con
el
ordenador

**Por
lotes:**
Las
informaciones
de
entrada
se
reúnen
en
un
punto
de
recepción
posteriormente
se
procesan
todas
juntas.

De
cualquier
forma
hay
que
definir:

Los
formatos
de
cada
una
de
las
pantallas
utilizadas

El
encadenamiento
entre
pantallas

Los
menús
o
diálogos

concretos
en
función
de
los
grupos
de
usuarios
que
los
utilizen

Para
el
diseño
de
las
pantallas
hemos
de
tener
en
cuenta
las
siguientes
consideraciones

Características
deseadas:
Simplicidad,
claridad
y
fácil
de
comprender

Saber
donde
situar
la
información
de
la
pantalla

Saber
qué
información
situar
en
la
pantalla

Saber
cómo
situar
la
información
en
la
pantalla

La
interfaz
de
entrada
debe
estar
protegida
contra
errores,
es
decir,
debe
recoger
todos
los
datos
necesarios
sin
introducir
errores
en
el
sistema

La
interfaz
de
salida
debe
asegurar
que
se
extraen
todos
los
datos
y
que
están
estructurados
de
forma
que

sean
fáciles
de
comprender.

Hacer
un
buen
usus
del
color
para
que
atraiga
la
atención
del
usuario.

...

M.P

M.C.E

M.C.T

M.C.S

b

a